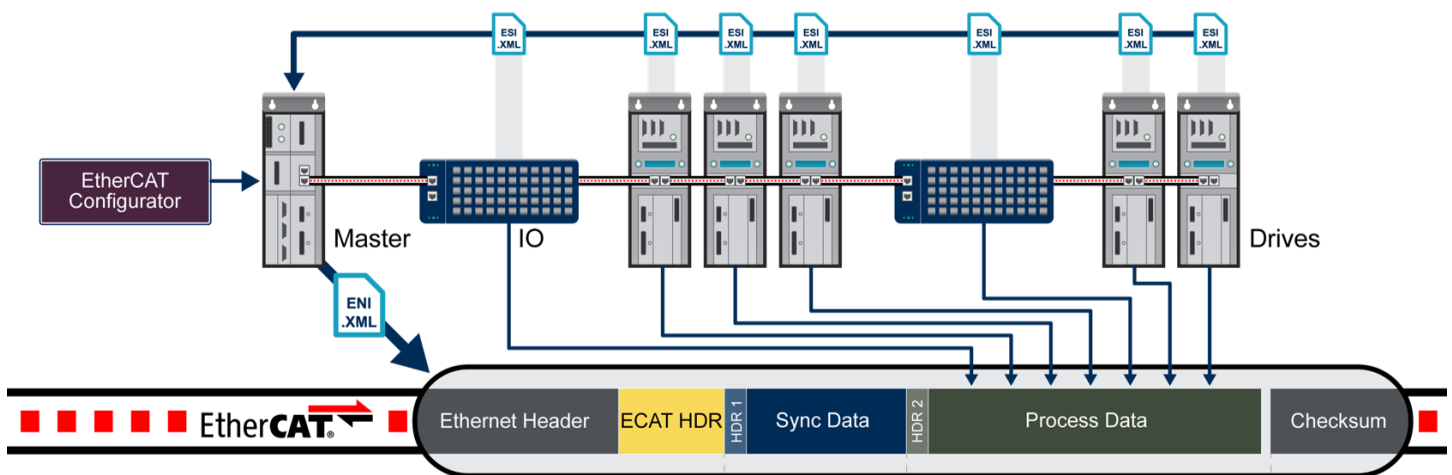


Auto-Configuration

Separating development and deployment: configuration strategies

When a machine is created both the software components and hardware components must be made and adjusted together. But to reduce development time and effort modules of hardware or software are made and re-used in different machines. Therefore, there is a development stage when the modules are created and a deployment stage when they are combined to create the machine. Technologies have to allow this flexibility, with EtherCAT this is the role of the ENI file.

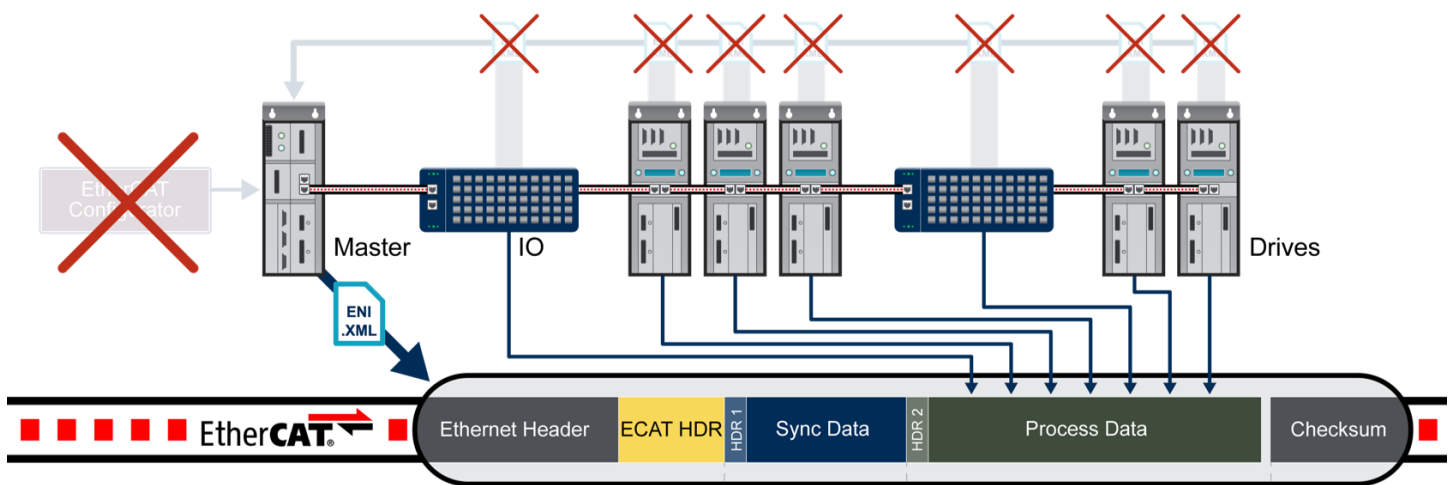


When using EtherCAT the program is developed based on variable names or locations. An EtherCAT Network Information (ENI) file describes how to configure the hardware for the right variables to be at the right location with the right name. When building a machine an EtherCAT Configurator software is used to create a configuration ENI file based on the functions and variables available in the hardware. This gives machine builders complete flexibility over which type of hardware is used with an application. However, this flexibility has a major drawback, the person or team assembling the machine has to understand the



hardware capabilities, the software requirements and the EtherCAT technology. If the configuration is incorrect the performance of the machine can be impacted greatly.

For companies which build controllers or modular machines this may be a major issue as it puts a lot of burden on the team deploying the systems. To reduce this burden most EtherCAT master system compete on how user-friendly their EtherCAT Configurator is. But most machines do not need that much flexibility. In most case each module will have mostly fixed hardware. Differences may be updates, low to high-end versions and optional components. In these cases, it is possible to give up part of the flexibility to remove completely the need to use an EtherCAT Configurator, this is called automatic configuration. When using automatic configuration, the EtherCAT master scans the bus and configures the hardware automatically. This information is given to the software application. The software application is developed to recognize modules and control them.



Using Automatic configuration, the development team no longer needs to define the correct configuration using a tool. All they have to do is to set the identification on the hardware for the software to uniquely identify them. To be able to create a configuration automatically the EtherCAT master has to know what features each hardware has and which of these features are needed by the application. Knowing the features available in the slaves is done using a hardware database, knowing the requirements of the application is done using APIs. This way the software application can directly request features and check what hardware is present and if the requested features are available.

As mentioned before there is a limitation with automatic configuration, the assumption that all identical hardware will use the same features in an application. This is not a very strict limitation as features can be enabled even if they are not used.

