

WHITE PAPER**Upgrading Bosch's E-Ray FlexRay IP-Module for Pretended Networking Support  
- Proposal for a Hardware Implementation**

Christian Horst, Robert Bosch GmbH, 72703 Reutlingen, Germany  
Christoph Schmutzler, Daimler AG, 71059 Sindelfingen, Germany

**Abstract**

**This White Paper describes an approach how to upgrade the existing and widely used E-Ray, the FlexRay IP-module developed by Robert Bosch GmbH, in order to support Pretended Networking as currently under definition in AUTOSAR. In a first step the E-Ray FlexRay Communication Controller [1] and the Intelligent Communication Controller [2, 7] were implemented as separate modules on an FPGA to demonstrate Pretended Networking for FlexRay. The next step, described in this White Paper, will be to integrate the Pretended Networking support of the Intelligent Communication Controller into the E-Ray IP-module. This avoids the disadvantages of separate implementations, like double storage of messages and extra effort to configure and control both components. This paper presents the enhancements with respect to the module structure and the impact on the E-Ray internal FlexRay protocol state machine.**

**1. Introduction**

The AUTOSAR concept Pretended Networking aims at providing means for a demand-dependent ECU-selective deactivation during active bus communication. This is currently not supported by FlexRay.

An *Intelligent Communication Controller (ICOM)* can be seen as a functional extension of a *FlexRay Communication Controller (CC)*. In particular, an ICOM

- filters bus traffic for configurable wakeup conditions,
- performs timeout monitoring on configurable frames,
- optionally continues to send static frames with a given cycle time,
- and buffers configured frames that contain data which is required after the ECU resumes normal operation.

This allows ECUs to go to sleep when they are temporarily not required. Once the ICOM detects a wakeup event, the ECU resumes its operation.

In the expected form for AUTOSAR R4.0.4, Pretended Networking only provides an intermediate solution for CAN that does not include the state machines of the affected software modules of the AUTOSAR communication stack. For future AUTOSAR releases, Pretended Networking is planned to be extended and also include support for FlexRay ICOMs. Please note, that the ICOM features described in this paper should be considered as a hardware implementation proposal and are not specified in AUTOSAR.

This paper provides an overview of a possible E-Ray implementation that includes the ICOM features for FlexRay. We propose to merge the ICOM features into the existing E-Ray FlexRay IP-module instead of developing a separate device. This reduces resource consumption and increases user friendliness of the ICOM features significantly. It is structured as follows. In Section 2, the general idea of an ICOM is explained. In Section 3, we introduce the features of the current E-Ray module and review the FPGA-based ICOM prototype. In Section 4, we outline the steps necessary to merge the ICOM features into a future E-Ray module that supports Pretended Networking and give a detailed description of such a module's structure, behaviour and required configuration. In Section 5, we conclude with a summary of next steps required to put these ideas into practice.

## 2. Intelligent Communication Controller

The development of the E-Ray IP-module at Bosch started in 2002. Revision 1.0 was conformance tested according to FlexRay Protocol V2.1 and delivered to the E-Ray licensees in May 2006. Up to now seven semiconductor manufacturers have obtained a license and integrated the IP-module into their microcontrollers. In addition, the E-Ray IP-module is implemented on FPGA, e.g. for use in FlexRay bus analyzer hardware.

Work on the ICOM concept started at Daimler in 2010 based on initial CAN-based ideas from Volkswagen. The ICOM concept had three main goals:

- allow a significant amount of modules of a FlexRay-enabled microcontroller, in particular the CPU and memories, to “sleep”, while the bus remains active,
- avoid timeout errors in other nodes by continuing to send static status frames,
- and provide a mechanism to wakeup a sleeping microcontroller based on bus traffic, e.g. by monitoring relevant signals for wakeup reasons or when detecting an error.

This would allow an ECU to go to sleep in intervals where its functionality is not required.

Take, for example, the parking assist ECU that provides feedback regarding the remaining space around the car up to a speed of 20 kph. Above that speed, most internal components can be disabled, but the microcontroller always remains powered because of ongoing bus communication. Using an ICOM, the microcontroller could go to sleep as well. Wakeup conditions are vehicle speed and the ignition state. Please note that the bus driver needs to be in state “normal” when the ICOM is active.

An even greater amount of ECUs could be disabled during the recharge of the high-voltage battery of hybrid or all-electric vehicles. The recharging process requires cyclic communication only between few selected ECUs. Without ICOMs all FlexRay ECUs on the same bus would be active, however. With ICOMs, ECUs that are not required could go to sleep during the charging process.

Most microcontrollers with an integrated FlexRay Communication Controller are typically high-performance devices with an average supply current of 200 mA up to 400 mA. Thus we expect valuable energy savings with positive effects on CO<sub>2</sub> emissions and charging efficiency, even though the ICOM and the bus driver must still be powered.

Safety and reliability concerns are addressed by defining appropriate ICOM configurations that consider all possible wakeup events. Using protocols defined for Partial Networking, other nodes can be informed about frames that are not sent during sleep.

The next section provides a detailed description of the hardware implementation approach used to verify the ICOM concept.

## 3. Current Design

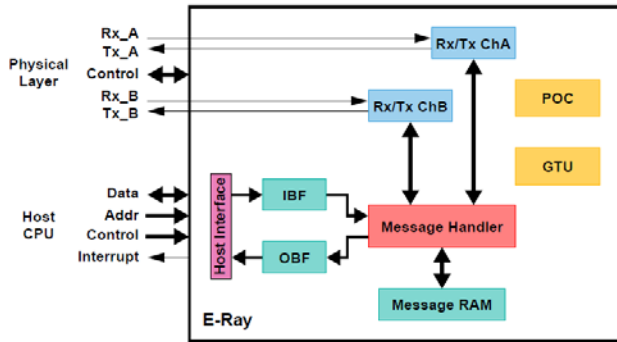
The prototypical ICOM implementation presented in this work can be seen as a functional extension of the FlexRay Communication Controller. The Communication Controller itself is not modified, all ICOM features are implemented in a separate module.

### 3.1 E-Ray

The E-Ray FlexRay Communication Controller supports FlexRay protocol V2.1 with data rates of up to 10 MBit/s on each channel. Up to 128 message buffers are available for communication. They can be configured for payload lengths of up to 254 bytes. A number of message buffers may be concatenated to form a receive FIFO. The E-Ray supports filtering of messages based on cycle counter, slot counter, and channel.

The Host CPU can access the E-Ray’s internal message storage via Input Buffer (IBF) and Output Buffer (OBF). The Message Handler controls the data transfer between the Channel Logic (Rx/Tx ChA,B) and the Message RAM, as well as the data transfer from/to the Host CPU via IBF and OBF. This mechanism

resolves access conflicts to the message buffers located in the Message RAM between Host CPU and Channel Logic and thereby guarantees data consistency.

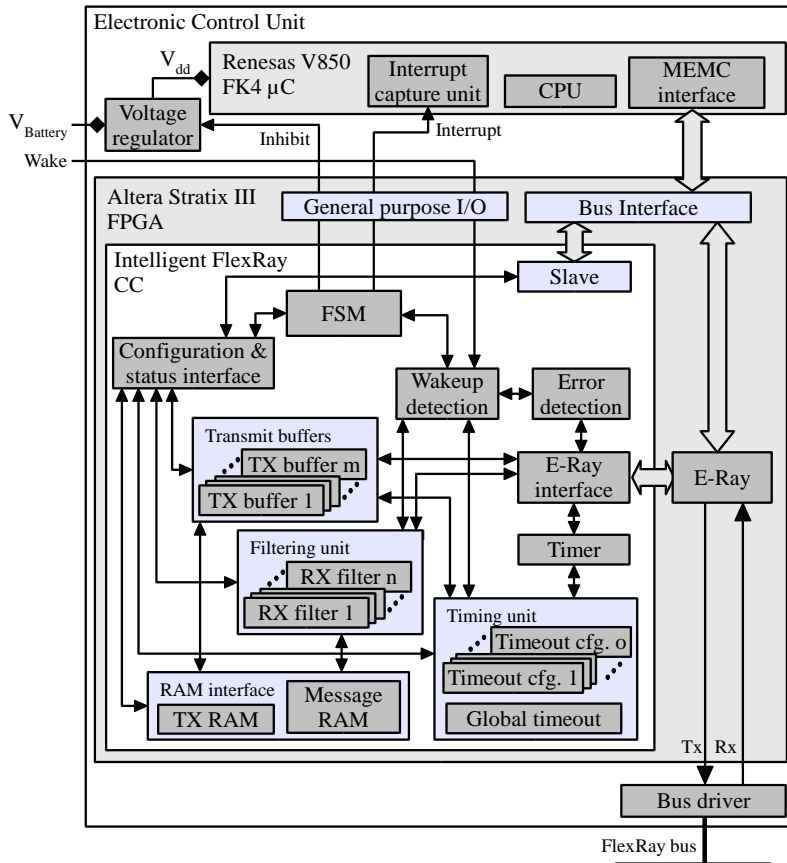


**Figure 1 Block diagram E-Ray IP-module.**

The FlexRay protocol states are handled by the Protocol Operation Control (POC) while the Global Time Unit (GTU) does the fault tolerant clock synchronization and controls slot and cycle counter. The Message RAM stores the message buffers together with the related configuration data.

### 3.2 ICOM

To validate the ICOM concept, we developed a test platform that allows us to use our FPGA-based ICOM implementation with a V850E2/FK4 microcontroller from Renesas Electronics [6]. The FK4 supports an external memory controller (MEMC) interface that can be accessed via a designated address space.



**Figure 2 Prototypical FPGA-based ICOM implementation.**

As shown in Figure 2, the FK4 MEMC interface is connected to an Altera Stratix III FPGA (EP3SL150F1152C2). A custom wrapper block is used to connect the MEMC interface to the FPGA's on-

chip bus. This allows us to directly access IP blocks on the FPGA from the FK4. From a software perspective, it is not noticeable whether a component is located on the microcontroller or on the FPGA. This setup allows to integrate the ICOM into an AUTOSAR stack and into existing application software. It also allows us to measure and evaluate performance figures such as reaction times or required software resources for ICOM configuration and management using the target hardware.

The main ICOM components are the Filtering and Timing units, the Transmit buffers and the RAM interface to an internal memory. The memory contains the configuration of TX buffers, the RX and timeout filters. It is also used to store messages that have passed the RX filters. The number of TX buffers, RX and timeout filters are configurable.

By storing frames that matched an RX filter, wakeup reaction time can be significantly reduced. After a wakeup, the software can read back frames that are required to resume normal operation without the need to wait for the next transmission.

The internal memory can be accessed by the ICOM Status interface to read back stored messages after a wakeup. The Configuration interface encapsulates the settings of the different ICOM components.

For communication, the ICOM uses the E-Ray IP-module. The E-Ray is also used by the CPU during normal operation. The ICOM's E-Ray interface polls the E-Ray for new frames and stores received frames in an internal buffer. It then notifies the Filtering unit about the received frame.

For each new frame, the Filtering unit loops through all filter configurations. If the received frame matches the configured frame ID, it compares a segment of the payload with the configured compare value by applying an arithmetic compare operation. If the result is positive, the frame is stored in the ICOM's Message RAM and the Timeout unit is notified of the matching filter. Depending on the configuration, the Filtering unit also signals when a wakeup event occurred.

The configuration of an RX filter is defined by the following parameters:

- a target E-Ray message buffer that is configured to receive frames (defined by slot, cycle offset and cycle repetition).
- a bit range within the payload that is
  - either compared to a static value with a given arithmetic operation (i.e., <, <=, =, !=, >=, >),
  - or checked for a change of value between two consecutive receptions, ignoring the actual value itself.
- an option to only execute the filter if a single bit at a given position in the payload has the value "1" (e.g. to check for PDU update bits at the beginning of the payload).
- an option to store the received frame in the E-Ray's Message RAM if the filter has matched.
- an option to issue a wakeup event if the filter has matched.

The Timeout unit can be configured to issue a wakeup if certain RX filters have not matched for a configurable amount of time. Therefore, the Timeout unit can be used for timeout monitoring of signals, or to emulate AUTOSARs network management behaviour.

The ICOM's TX buffers allow to autonomously send frames with a static payload. Each TX buffer provides the following configuration options:

- a target E-Ray message buffer that is configured to send frames,
- the static payload that shall be sent when the ICOM is active,
- and an arbitrary cycle time that is not bound to the slot and cycle configuration of the E-Ray message buffer.

If the cycle time of a buffer has expired, it writes its payload into the E-Ray, requests the transmission of the frame via the E-Ray interface and resets the cycle time.

The Wakeup detection raises a wakeup event on a matching RX filter or timeout event. It also monitors a dedicated Wake-pin that can be used to generate local wakeup events (e.g. press of a button). The Error detection block observes the E-Ray's POC-status [1].

A Finite State Machine (FSM) controls the Interrupt and Inhibit pin based on the current ICOM state and wakeup events. The Inhibit pin is set low in Pretended Networking mode to turn off the  $\mu\text{C}$ 's supply voltage. The Interrupt signal is activated when a wakeup, time out or error event occurred. Since all components require the internal ICOM state, the FSM is connected to all other blocks (omitted from Figure 2).

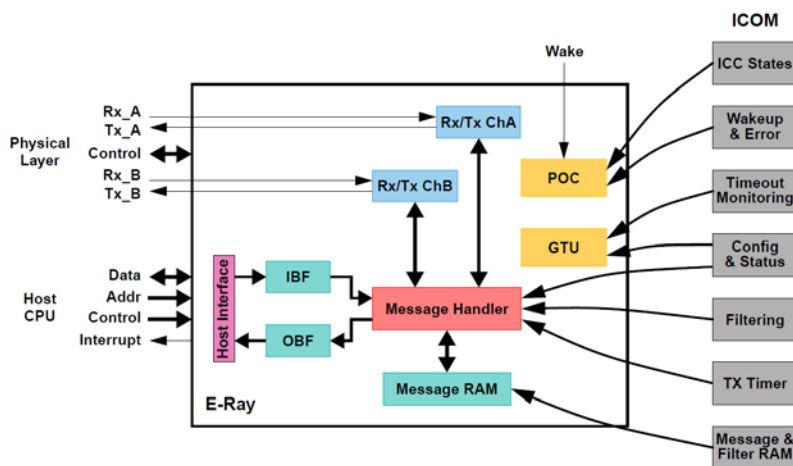
The separate implementation of a FlexRay Communication Controller and an Intelligent Communication Controller for Pretended Networking has the disadvantage that components like message storage and filter mechanisms are implemented twice. For receive filtering, all received messages are transferred from the E-Ray to the Intelligent Communication Controller to check for wakeup messages. This may lead to loss of messages if timing requirements are violated. In addition configuration and control of both components by software requires extra effort and care. Both components are powered when the FlexRay node is in Pretended Networking operation.

Therefore, we expect that in future implementations the ICOM functionality will be integrated into the E-Ray. We also expect the E-Ray to be integrated in a microcontroller and not to be realized as a separate ASIC. In this case, the E-Ray should be realized as a separate power domain (e.g. by power or clock gating mechanisms), so that during Pretended Networking all microcontroller components except the E-Ray could be put into an energy efficient state. Apart from the bus driver and the microcontroller's voltage supply, the remaining parts of the ECU can be powered off.

#### 4. Integration of ICOM Functionality into the E-Ray Design

The ICOM features could reuse the existing E-Ray configuration and resources to a great extent when being integrated. In the following sections, we outline which E-Ray modules are affected by an integration of the ICOM, how the E-Ray's state machine could be extended to incorporate the new communication behaviour, and which additional configuration parameters are required.

##### 4.1 Structure



**Figure 3 Merge of ICOM functions into E-Ray IP-module.**

As shown in Figure 3, the following parts of the E-Ray implementation are affected by the integration of an Intelligent Communication Controller:

- an additional state has to be added to the FlexRay Protocol state machine (POC).
- a Wake input pin is added to support wakeup by an external event.
- the Global Time Unit (GTU) has to be enhanced to check for timeout conditions.
- the Message Handler has to be enhanced to support filtering for wakeup conditions.
- a TX timer unit has to be added to trigger autonomous message transmission when the E-Ray is in state ICOM ON and the Host is shut down.
- the Message RAM has to be expanded to store the filter configurations and to store frames that have passed the receive filters.

- an interrupt output signal has to be added to signal wakeup or error conditions to the Host CPU.

After the merge the E-Ray supports Pretended Networking functions as planned in a future AUTOSAR release.

#### 4.2 State Machine

To integrate the ICOM functionalities into the E-Ray, its POC state machine is extended by a new state **ICOM ON**, as shown in Figure 4. **ICOM ON** can be seen as an add-on to the FlexRay protocol state machine, the FlexRay protocol and conformance are not affected. Table 1 lists the state transitions for the new ICOM state added to the E-Ray overall state machine. All other state transitions are described in [1].

ICOM relevant filter and timeout configurations can be changed in every state except for **ICOM ON**. The message buffers are accessible in all states. In **NORMAL ACTIVE**, the Host can request a transition to **ICOM ON**. At the end of the current communication cycle, the E-Ray switches to **ICOM ON** and then filters incoming messages according to its configuration. In case of a wakeup event (matching filter, timeout condition, or event on pin Wake), it switches back to **NORMAL ACTIVE**, stops filtering of incoming messages and raises an interrupt. During **ICOM ON**, the Host can also trigger a transition to **NORMAL ACTIVE**, **READY**, or **HALT** state. In these cases, no interrupt is raised.

In case of a communication related error, the E-Ray switches to **NORMAL PASSIVE** or **HALT**. The transitions are signaled by an interrupt and are triggered by the same criteria as when transitioning to **NORMAL PASSIVE** or **HALT** from **NORMAL ACTIVE**. After the transition the potential error condition must be handled by software.

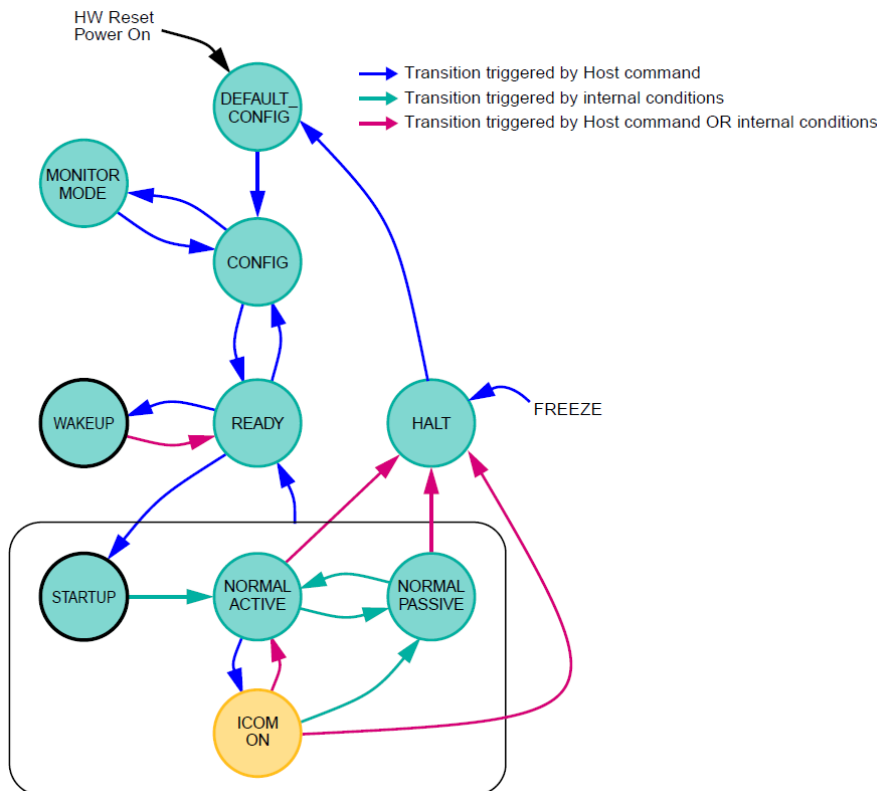


Figure 4 Enhanced state diagram of the E-Ray's POC state machine.

Condition	From	To	IRQ raised
Host command ICOM ON	NORMAL ACTIVE	ICOM ON	No
Host command NORMAL ACTIVE	ICOM ON	NORMAL ACTIVE	No
ICOM wakeup condition occurred (matching filter, timeout condition, or event on pin Wake)	ICOM ON	NORMAL ACTIVE	Yes
Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit	ICOM ON	NORMAL PASSIVE	Yes
Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit AND Halt due to Clock Sync Error enabled	ICOM ON	HALT	Yes
Host command READY	ICOM ON	READY	No
Host commands HALT, FREEZE	ICOM ON	HALT	No

**Table 1 Additional ICOM state transitions of the E-Ray's overall state machine.**

### 4.3 Error Handling

From an applications point of view, the following errors can occur during **ICOM ON**:

- Sync loss of the E-Ray (e.g. caused by bus glitches or an erroneous bus driver, cf. [3]) causing the E-Ray to go to **HALT**.
- Timing problems causing the E-Ray to go into **NORMAL PASSIVE** (cf. [3]).
- Absence of frames that are used in the ICOM filtering configuration, resulting in a timeout condition (e.g. caused by a malfunction of the sending node).

A sync loss or timing errors are already detected by the existing E-Ray IP-module. By raising an interrupt triggered by a transition from **ICOM ON** to **NORMAL PASSIVE** or **HALT** caused by an error, we can ensure that error handling can be done in software. To cope with errors of other nodes, the E-Ray is extended by a timeout detection ability: RX filters may trigger a wakeup event if they did not match for a configurable amount of time. Using this mechanism, the E-Ray is able to detect missing frames or signals that are not updated by the sender. After the E-Ray triggered a wakeup, further error handling must be done in software.

### 4.4 Configuration

The main add-on to the E-Ray's functionality is the ability to autonomously

1. filter incoming frames for wakeup conditions (i.e., signal values),
2. perform timeout monitoring of filters,
3. store relevant frames in an internal memory,
4. and send frames with a static payload with a given cycle time.

The behaviour of the send buffers, receive and timeout filters is configurable and must be set before switching to **ICOM ON**. The configuration could be based on the parameters described in Section 3.2.

## 5. Conclusion

As shown above, the integration of an Intelligent Communication Controller that supports Pretended Networking into a FlexRay Communication Controller has significant advantages with respect to resource requirements (logic area and RAM size) and ease of use. The FlexRay protocol functions as specified in the FlexRay Protocol Specification V2.1 / V3.0.1 or ISO17458 are not affected by this add-on. The E-Ray FlexRay IP-module with Pretended Networking support may be integrated as stand-alone device or as microcontroller peripheral. In case of a microcontroller peripheral, a separate power-domain for the E-Ray and the related clock generation would be beneficial to optimize power consumption.

Depending on demand for FlexRay Pretended Networking, Bosch is planning to upgrade their E-Ray FlexRay IP-module to support Pretended Networking as described in this paper, and according to the requirements of AUTOSAR. This could be done in line with an upgrade to FlexRay V3.0.1 resp. ISO17458.

## References

- [1] Robert Bosch GmbH, E-Ray FlexRay IP-Module User's Manual Revision 1.2.7, Available: [www.semiconductors.bosch.de/media/en/pdf/ipmodules\\_1/flexray/eray\\_users\\_manual\\_1\\_2\\_7.pdf](http://www.semiconductors.bosch.de/media/en/pdf/ipmodules_1/flexray/eray_users_manual_1_2_7.pdf)
- [2] C. Schmutzler, A. Lakhtel, M. Simons, and J. Becker, "Increasing Energy Efficiency of Automotive E/E-Architectures with Intelligent FlexRay Communication Controllers," System on Chip (SoC), 2011 International Symposium on, Nov. 2011
- [3] FlexRay Consortium, FlexRay Communication System Protocol Specification V2.1 Rev A, Available: [www.flexray.com](http://www.flexray.com)
- [4] FlexRay Consortium, FlexRay Communication System Protocol Specification V2.1 Rev A Errata V1, Available: [www.flexray.com](http://www.flexray.com)
- [5] FlexRay Consortium, FlexRay Communication System Protocol Specification V3.0.1, Available: [www.flexray.com](http://www.flexray.com)
- [6] V850E2/Fx4 Microcontrollers, Renesas Electronics Corporation, Available: [www.renesas.com/products/mpumcu/v850/v850e2fx/v850e2fx4/index.jsp](http://www.renesas.com/products/mpumcu/v850/v850e2fx/v850e2fx4/index.jsp)
- [7] C. Schmutzler, M. Simons, and J. Becker, "On Demand Dependent Deactivation of Automotive ECUs", DATE2012, March 2012