ANALOG DEVICES

Revolutionizing IIoT: Enabling Real-time Secure Connectivity to the Edge with Single-Pair Ethernet and Zephyr® OS

Jason Murphy
Senior Engineer, Software Systems

Zephyr® Project
Developer Summit

analog.com

Agenda

- Intro to 10BASE-T1L Single-Pair Ethernet
- The Need for Software
- Zephyr for SPE Application Development
- Summary

Ethernet to the Edge

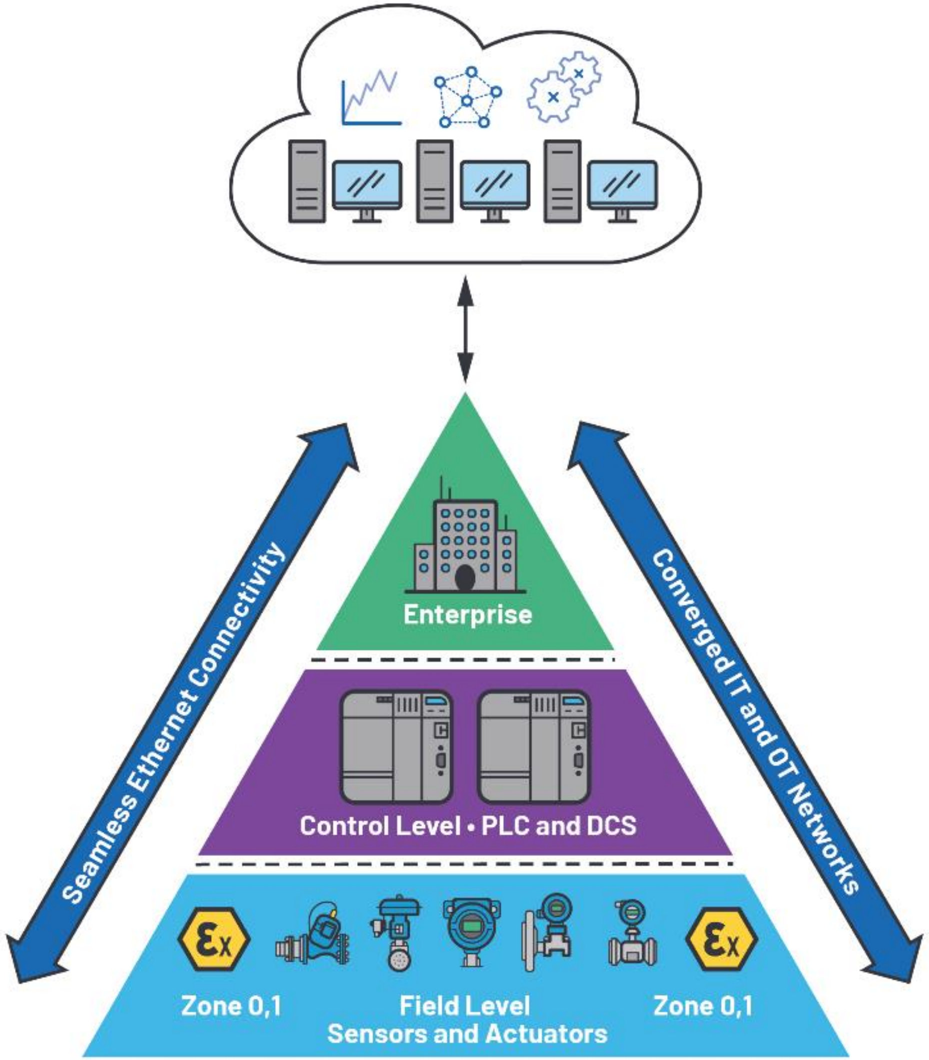
Industrial connectivity networks have adopted Ethernet over the last number of decades.

Ethernet has been adopted where possible in industrial networks (Control Level).

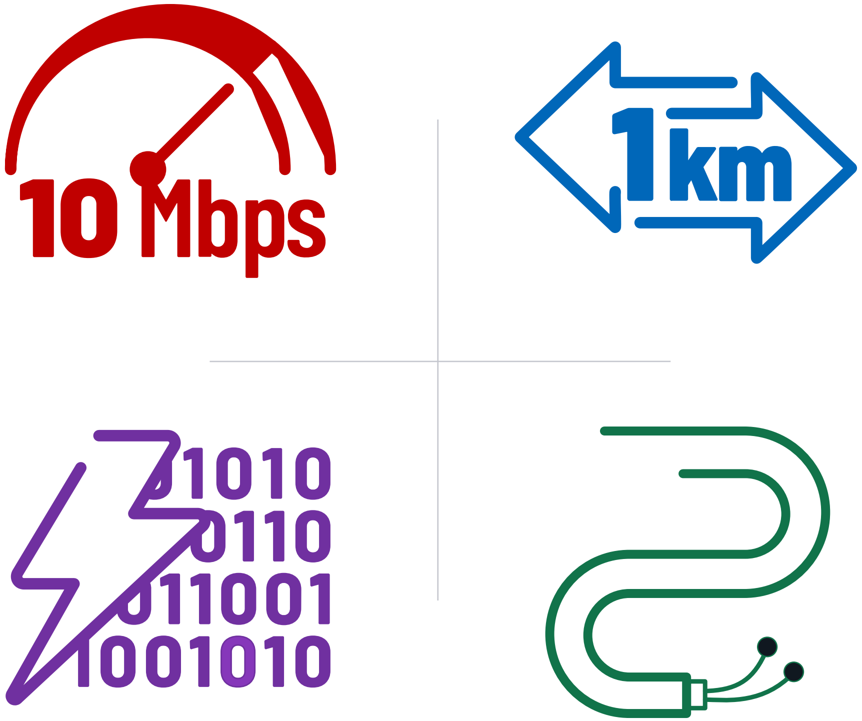
Gap: Ethernet to the Edge

IEEE 802.3cg-2019 (10BASE-T1L)

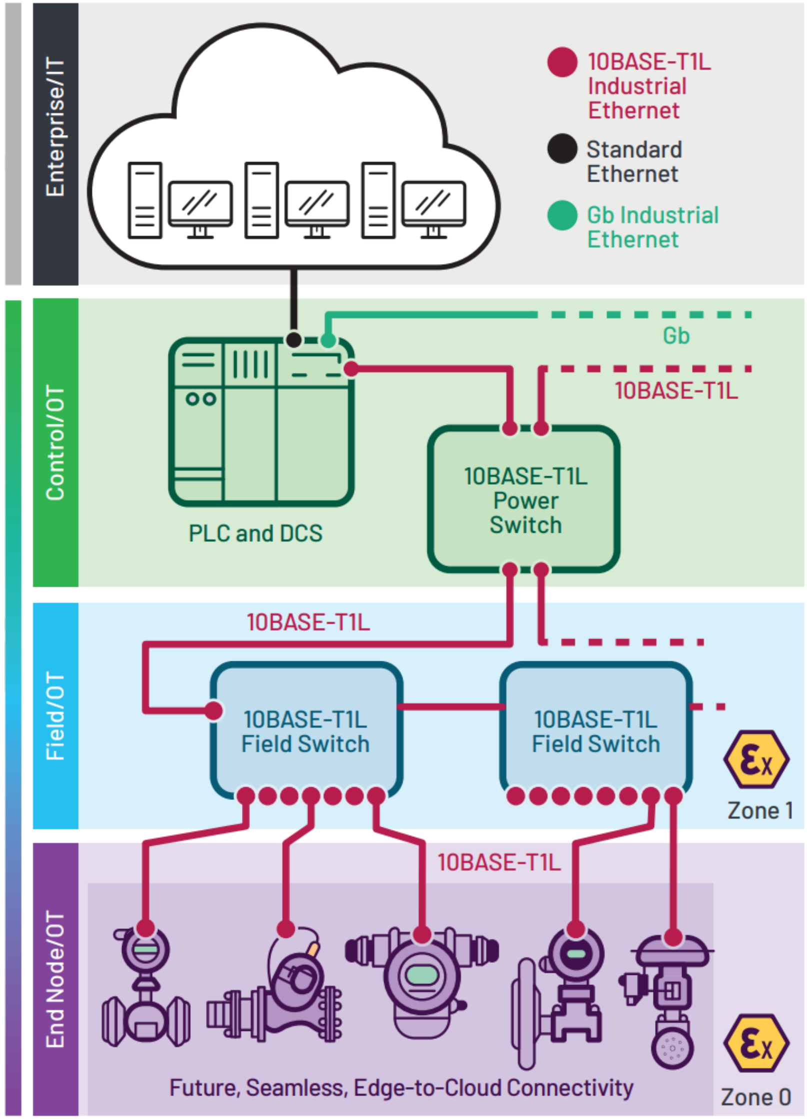
Enables Ethernet connectivity to field-level/edge devices.


Enterprise
Control Level • PLC and DCS
Zone 0,1
Field Level
Sensors and Actuators
Zone 0,1
Seamless Ethernet Connectivity
Converged IT and OT Networks

10BASE-T1L SPE Overview



10BASE-T1L in the Industrial Network



10BASE-T1L - Key Benefits

10 Mbps

Data Rate

Higher Speed vs. Legacy Fieldbus

Improved Latency

1km

IP to the Edge

Remove Data Islands

Remove Gateways

Simplify Data Access

Security

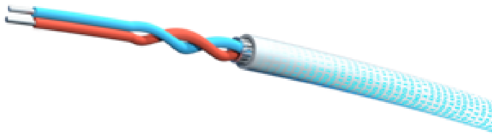
Difficult to Retrofit Security

Ethernet Enables Modern Security

Diagnostics

Send More Than Just Sensor Data

Send Device Fault & Diagnostics Data



10BASE-T1L Transceivers

PHY

- Host processor includes Ethernet MAC
- MDIO/MII interface
- High performance

MAC/PHY

- Host processor does not need Ethernet MAC
- SPI interface
- Suitable for Low-Power MCUs

2-PORT SWITCH

- 2-Port Switch (2x PHY + MAC)
- SPI Interface
- Enables various network topologies

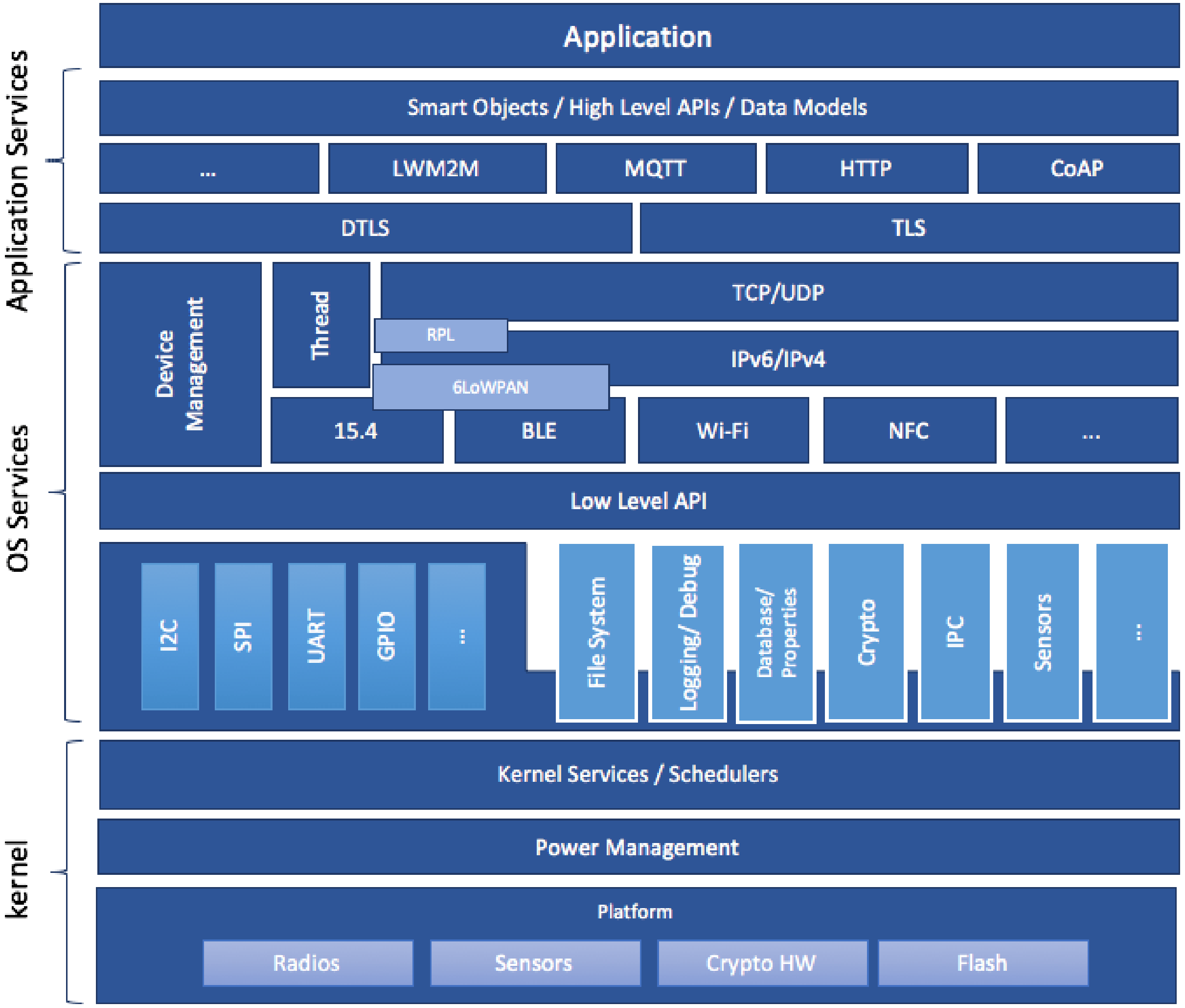
The Need for Software

Embedded Ethernet can be challenging.

Software is key!

- Device Drivers
- Connectivity Stack (TCP/IP)
- Application Protocols
- Security (Data & Device)
- Edge Processing

Why Zephyr?



Native TCP/IP Networking Stack

Support for key Ethernet protocols

Real-Time Performance

Deterministic, low latency networking

Resource Efficient

Optimised for resource-constrained devices

Security

Encryption, authentication, secure boot

Open Source

Transparent, extensible software

Portability

Support for a range of platforms and arch's

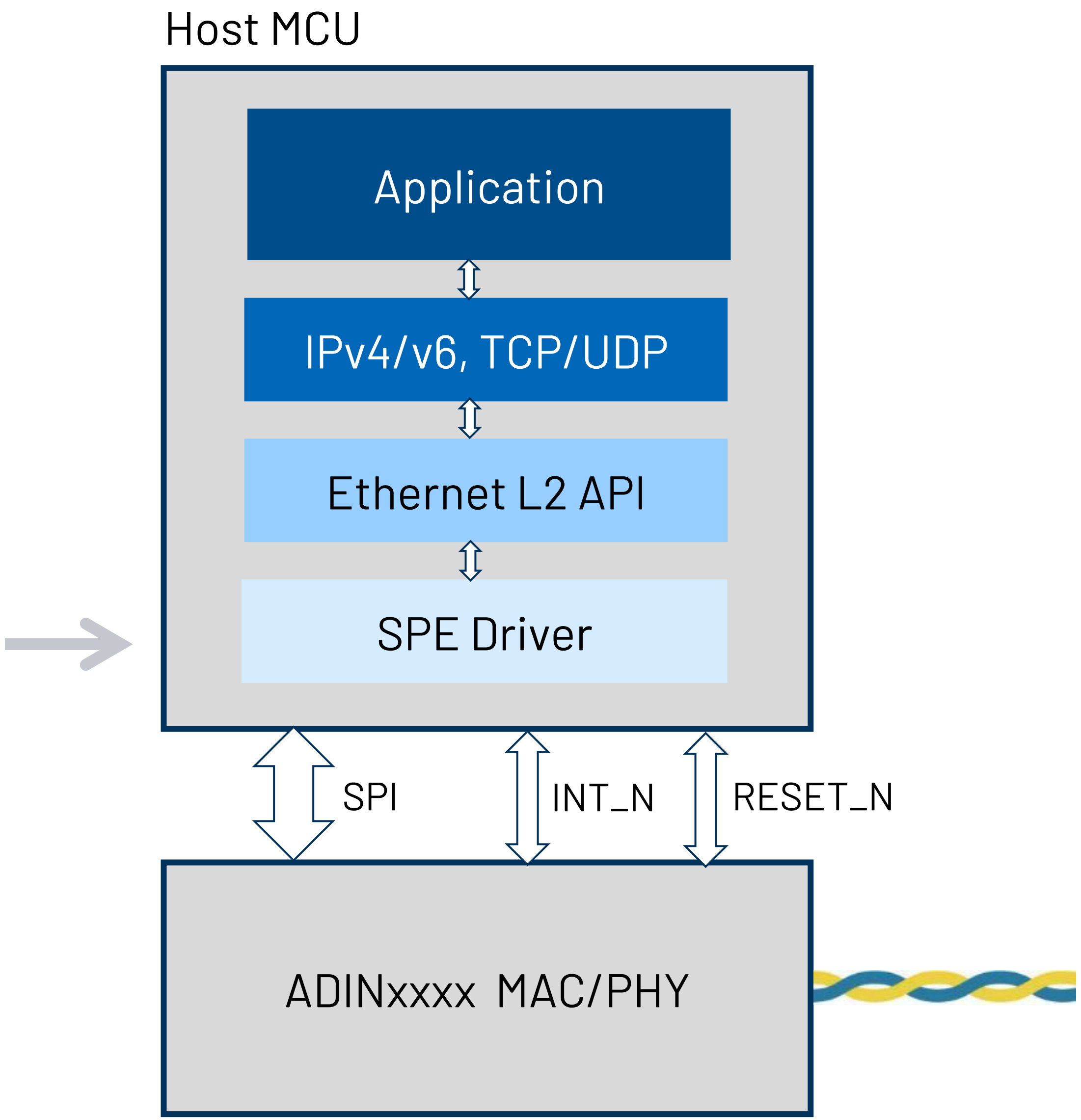
1) https://docs.zephyrproject.org/latest/security/security-overview.html

10BASE-T1L MAC/PHY Zephyr Driver

Driver acts as the interface between the Ethernet MAC/PHY and the Zephyr network stack.

Zephyr defines a standard L1 API for Ethernet devices.

The standard API abstracts the physical layer details from the upper network stack layers.



10BASE-T1L MAC/PHY Zephyr Driver: Driver API

```
struct ethernet_api {
        struct net_if_api iface_api;
        int (*start)(const struct device *dev);
        int (*stop)(const struct device *dev);
        enum ethernet_hw_caps (*get_capabilities)(const struct device *dev);
        int (*set_config)(const struct device *dev, ..);
        int (*send)(const struct device *dev, struct net_pkt *pkt);
};
```

```
static int adin2111_init(const struct device *dev)
{
    // Check SPI interface
    // Config RESET_N and INT_N GPIOs
    // Register interrupt callbacks
    // Soft reset
    // Create offload thread
}
```

10BASE-T1L MAC/PHY Zephyr Driver: Open Alliance SPI

OPEN Alliance 10BASE-T1x MAC-PHY Serial Interface
Available in Zephyr → drivers/ethernet/oa_tc6.c

Enables 10Mbps frame transfer with SPI clock of 12-16MHz

OPEN ALLIANCE[1]

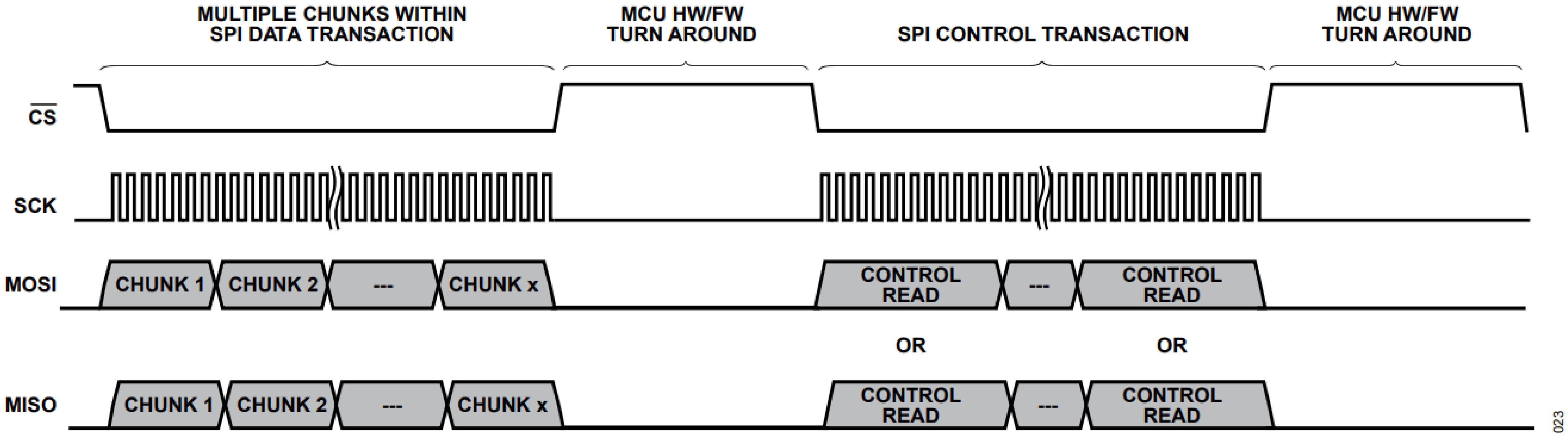
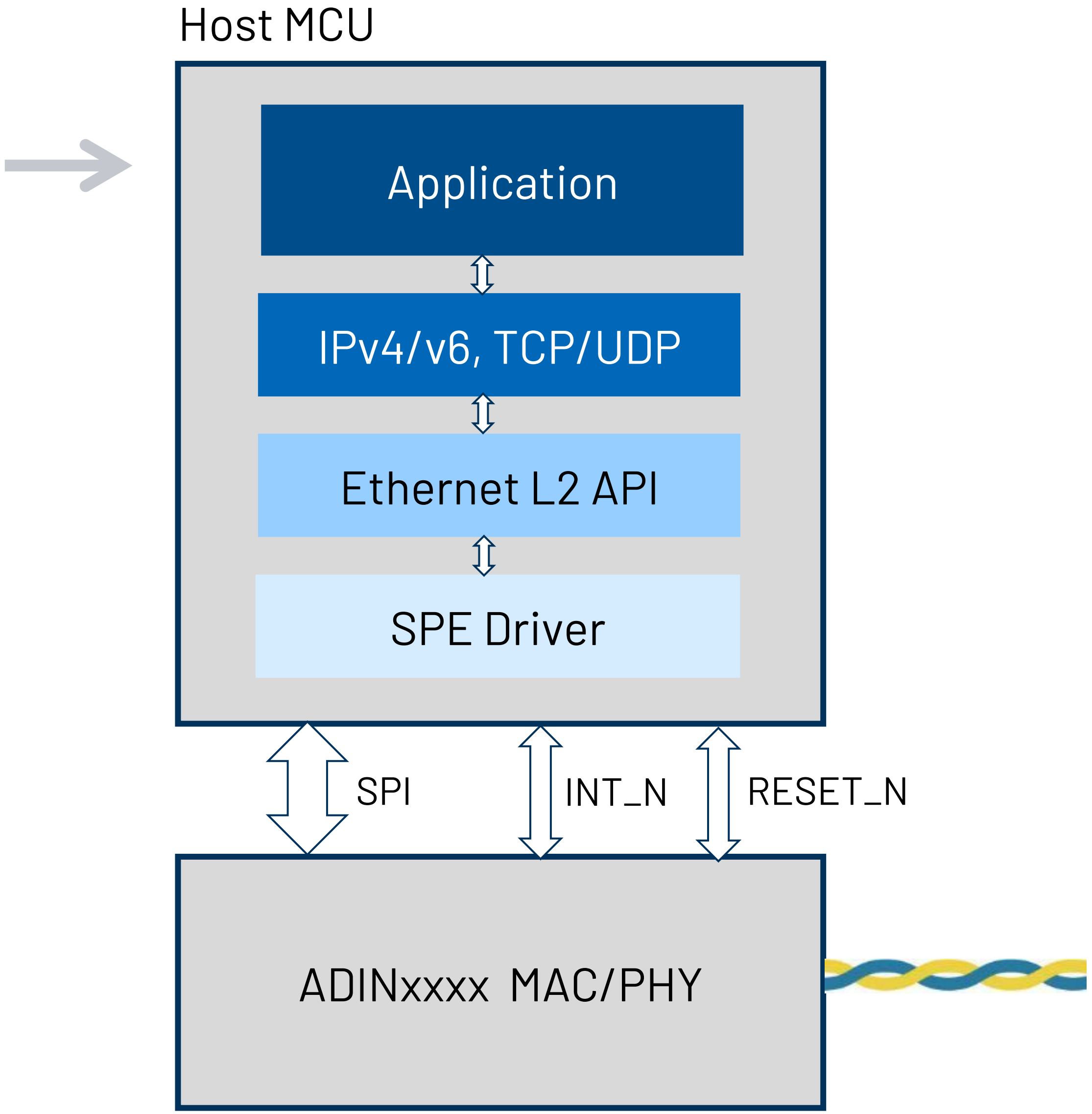


Figure 24. Ethernet Data Frame Transfer Followed by Control Transfer

[1] https://opensig.org/

Ethernet Application Development

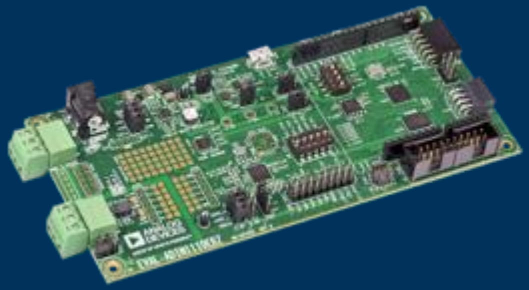
1. Platform Selection.
2. Board Configuration.
3. Software Configuration.
4. Application development using Zephyr networking libraries.



10BASE-T1L Platforms in Zephyr

ADIN1110 MAC/PHY

ADIN2111 2-Port Switch



adi_eval_adin1110ebz

adi_eval_adin2111ebz

Location in Zephyr tree:

- drivers/ethernet/eth_adin2111.c
- drivers/ethernet/phy/phy_adin2111.c

- boards/adi

Board Configuration - Devicetree

Devicetree configs:

- SPI bus/pins
- Reset/INT pins
- SPI frequency
- SPI DMA config
- MAC address
- Ethernet LED function

```
&spi3 {
    pinctrl-0 = <&spi3a_ss0_p0_19 &spi3a_miso_p0_20
                &spi3a_mosi_p0_21 &spi3a_sck_p0_16>;
    pinctrl-names = "default";
    status = "okay";

    adin1110: adin1110@0 {
        compatible = "adi,adin1110";
        reg = <0x0>;
        spi-max-frequency = <25000000>;
        int-gpios   = <&gpio0 17 (GPIO_ACTIVE_LOW | GPIO_PULL_UP)>;
        reset-gpios = <&gpio0 15 GPIO_ACTIVE_LOW>;
        port1 {
            local-mac-address = [ CA 2F B7 10 23 70 ];
        };
        mdio: mdio {
            compatible = "adi,adin2111-mdio";
            status = "okay";
            #address-cells = <1>;
            #size-cells = <0>;
            ethernet-phy@1 {
                reg = <0x1>;
                compatible = "adi,adin2111-phy";
                status = "okay";
                led0-en;
                led1-en;
            };
        };
    };
};
```

Software Configuration - Kconfig

Kconfig is used to enable the Ethernet driver and network stack components.

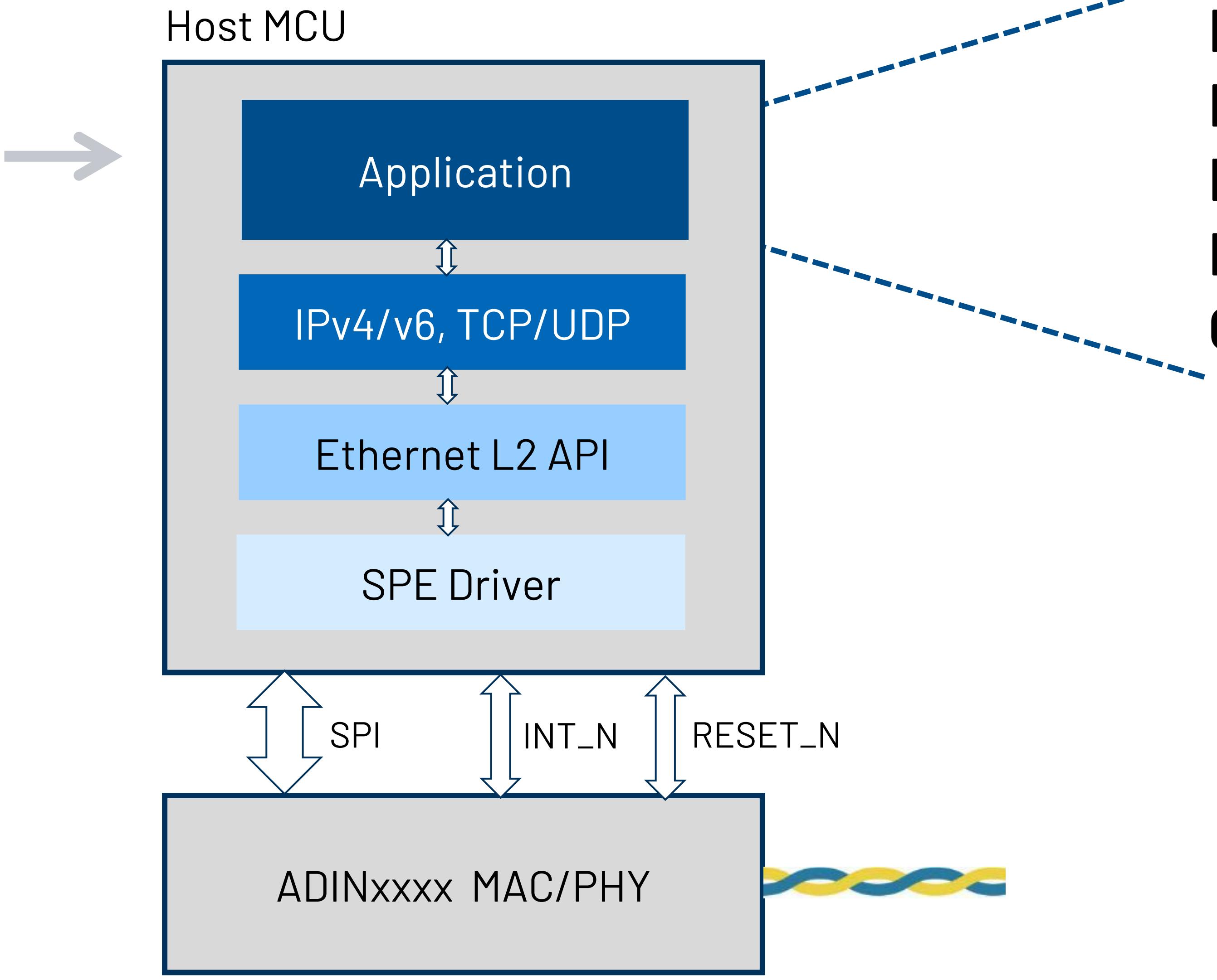
Software requirements will be application dependent.

Zephyr supports many L2/L3/L4 protocols:

L2:	L3:	L4:
LLDP	IPv4	TCP
	IPv6	UDP
	ICMP	
	ARP	

```
# Enable ADIN2111 driver
CONFIG_ETH_DRIVER=y
CONFIG_ETH_ADIN2111=y
CONFIG_NET_L2_ETHERNET=y

# Enable network stack
CONFIG_NETWORKING=y
CONFIG_NET_LOG=y

# Enable IPv4
CONFIG_NET_IPV4=y
CONFIG_NET_CONFIG_NEED_IPV4=y

# Enable TCP
CONFIG_NET_TCP=y

# Enable DHCP
CONFIG_NET_DHCPV4=y

# Enable Sockets (used by MQTT lib)
CONFIG_NET_SOCKETS=y
CONFIG_NET_SOCKETS_SOCKOPT_TLS=y

# Enable MQTT
CONFIG_MQTT_LIB=y
CONFIG_MQTT_LIB_TLS=y
```

Application Development



BSD Socket
MQTT
HTTP
LWM2M
CoAP

DNS
DHCP
JSON library
Network mgmt.
zperf

Secured Ethernet Connectivity

Sending data securely over Ethernet requires using TLS/DTLS.

Zephyr provides TLS support using mbedTLS.

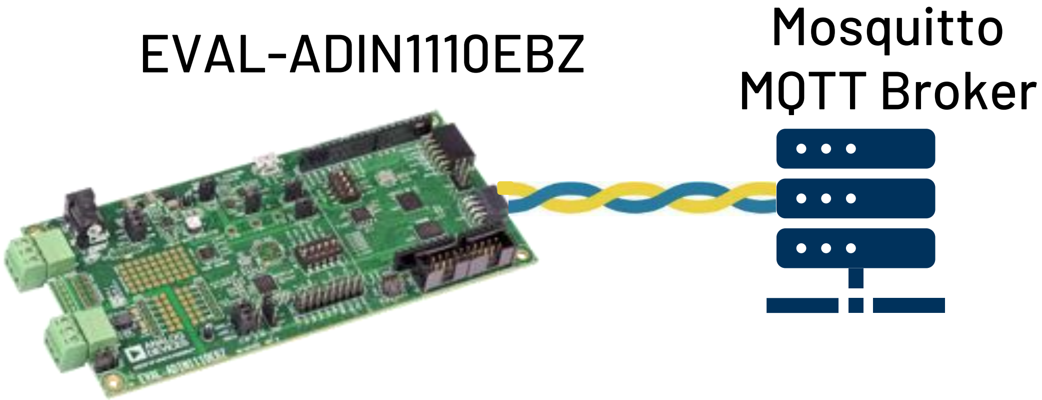
Zephyr’s Socket API includes secure socket capability.

Once a secure socket is configured, it can be used like a regular socket.

Enables protocols such as HTTP**S** and MQTT**S**

```
/* Register CA certificate */
tls_credential_add(CA_CERTIFICATE_TAG,
TLS_CREDENTIAL_CA_CERTIFICATE, ca_certificate,
sizeof(ca_certificate));

/* MQTT secure config */
client_ctx.transport.type = MQTT_TRANSPORT_SECURE;
struct mqtt_sec_config *tls_config =
&client_ctx.transport.tls.config;
tls_config->peer_verify = TLS_PEER_VERIFY_REQUIRED;
tls_config->cipher_list = NULL;
tls_config->sec_tag_list = m_sec_tags;
tls_config->sec_tag_count = ARRAY_SIZE(m_sec_tags);
tls_config->hostname = MQTT_BROKER_HOSTNAME;
```

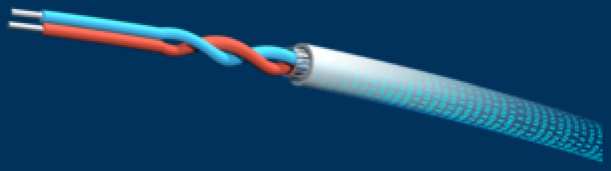
Sample Application



Features:

- Acquires IP address using DHCP lease.
- Brings up secure MQTT connection to broker using secure socket connection.
- Subscribes to user-defined MQTT topic(s).
- Work queue thread publishes temperature sensor data at a regular interval.
- Dedicated thread to handle incoming MQTT commands.

Summary



10BASE-T1L SPE is bringing Ethernet Connectivity to Edge Devices

Higher data rates over longer distance

Seamless connectivity from edge to cloud

Software is key to SPE device development

Embedded Ethernet requires software above L2

Zephyr provides an Embedded Software Ecosystem to build Ethernet-connected Edge Devices

Networking stack

Security

Real-time capabilities

SPE is already well supported in Zephyr

Provides an alternative to existing industrial comms protocols like CAN, Modbus while being able to use familiar socket/IP APIs.

1) https://www.zephyrproject.org/branding/

See our Secure Industrial Network Demo
@ ADI Booth (E15)

Thank you!