ams OSRAM Arduino OSP ecosystem evaluation kit

Quick Start Guide

AS1163_QF_EVM_KT_OSP

Published by ams-OSRAM AG
Tobelbader Strasse 30, 8141 Premstaetten, Austria
Phone +43 3136 500-0
ams-osram.com

Table of contents

1 Introduction

This document explains the key features of the **ams OSRAM Arduino OSP ecosystem evaluation kit** (abbreviated as EVK in this document).

The key devices of the EVK are the AS1163 (SAID) and the OSIRE® E3731i. They are OSP (Open System Protocol) compliant: They are able to communicate in a chain with up to 1000 nodes at 2.4 Mbit/s data rate, low overhead and fast-forwarding of messages. They can be configured in two different configurations: **bidirectional** and **loopback**. In addition, their design is specifically adapted for single layer PCBs or in applications where the space available forces the PCB to be reduced in dimensions. To be automotive compliant, the devices can bridge to the CAN physical layer if the signal must be driven outside of the PCB.

Lastly, the OSP is a complete license free protocol that is SPI like and thus compatible with many market MCUs. It is on purpose developed open source to give flexibility and freedom to the user.

The following chapters introduce OSP, the EVK as a whole and its boards in detail and conclude with introducing the accompanying software.

ams OSRAM developed a protocol to enable dynamic lighting applications, the Open System Protocol (abbreviated to OSP). As the name suggests, it is open, this means that everyone can access and use it. ams OSRAM decided to be flexible and compatible with other systems. This means not only that there are no barriers to other open systems but also the opportunity to receive full documentation and support without limitations and licenses. Everyone can participate in the development and growth of the protocol.

Any innovation comes from the partners and belongs to the partners!

1.1 Applications

The flexibility of OSP allows the implementation of Dynamic Lighting Applications for decoration, warnings, and communication, or even for the integration of interactive functions like in Smart Surfaces by the integration of sensors and actuators.

OSP supports dynamic lighting applications with the following key features:

- Address range: 1,024
- Net data: Up to 64-bit per message frame
- Data transfer rate: 2.4 Mbit/s
- Self-diagnosis, self-addressing, further network layer features

Figure 1: Example of dynamic light applications

The above pictures of Figure 1 represent four typical dynamic light applications, offering features like decorative, warning, communicating, and interacting. All these applications require individual control and addressability of hundred light points in one application. Light points are implemented by single color or multi-color LEDs. To allow a full color gamut, typically an RGB LED is used.

With OSP support integrated, each LED becomes intelligent, now including the individual characteristic, a temperature sensor and a serial bus interface. The master controller is now able without any further input to control and to address each of those devices in the chain. If the device is an intelligent RGB, the calibration can now be controlled by the master automatically and without any external input. In addition, it saves space. the serial bus reduces the number of connections to 4, independent of the number of connected devices.

1.2 Architecture

Originally implemented inside an RGB LED, the OSP offers advantages also in other devices like intelligent drivers for external LEDs. OSP is fast and flexible enough to allow these intelligent drivers to connect single or multi-color LEDs, other actuators or even sensors into the OSP network. This extends ideas of zone and domain architectures inside a car. The OEM can define its best architecture and just connects the "last mile" via OSP.

Figure 2: Decentralized brain example

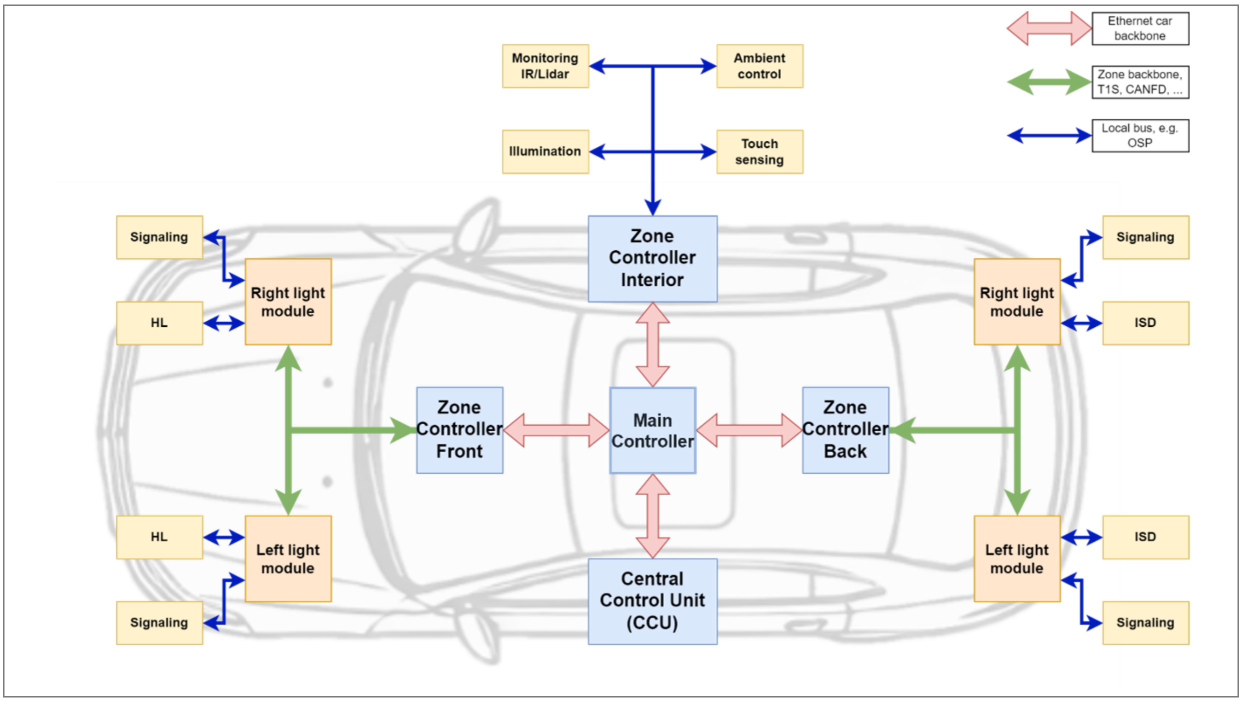
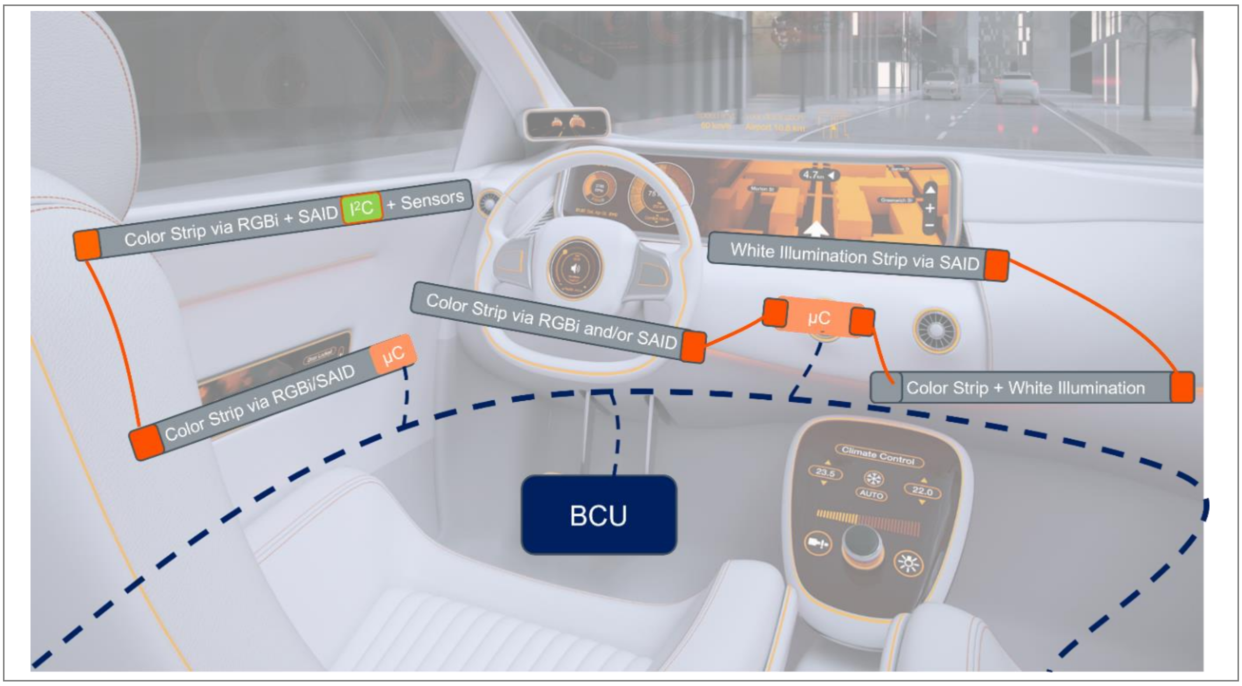


Figure 2 represents a Zone Controller Architecture. The OEM has defined its next generation of architecture layout, in this example using ETHERNET as backbone. The OSP network is colored blue in this picture. OSP connects to the backbone via Left/Right Light Modules, which operate as gateway. The Main Controller has, if necessary, complete control over the whole network. Alternatively, it can "decentralize" functionality into the zone or even the light modules itself. This flexibility may be used by the OEM to partition its network best.

Figure 3: Hypothetical use of OSP nodes



1.3 OSP devices ecosystem

AS1163 aka SAID and OSIRE® E3731i are the devices to explore in this EVK, shown in Figure 4 below. These smart nodes have been specifically designed to limit the complexity of the system. Having their own logic, the overall system can be reduced in terms of computational point of view of the central MCU. It is sufficient to just create and send the right message according to the protocol.

Figure 4: Left a strip with AS1163 and external RGB, right and OSIRE® E3731i device with integrated RGB

Traditionally, light emitters are devices without logic. In the past it was always necessary to generate a PWM signal to drive them. This would imply computational logic dedicated to the generation of the signal according to the board design too. OSP changes this. You have to construct a telegram following the OSP convention and send that to a node. The OSP devices and their internal PWM drivers will generate a constant PWM signal at constant current. Each device is completely responsible of understanding the command received and activate the proper internal block.

All the daisy chained devices are controlled by the Open System Protocol (OSP). It allows special messages to the nodes, which are smart enough to handle just a telegram and convert it to a command.

Interfacing the daisy chain of devices with just one microcontroller offers the user the possibility to exploit the potential of the smart devices AS1163 and OSIRE® E3731i.

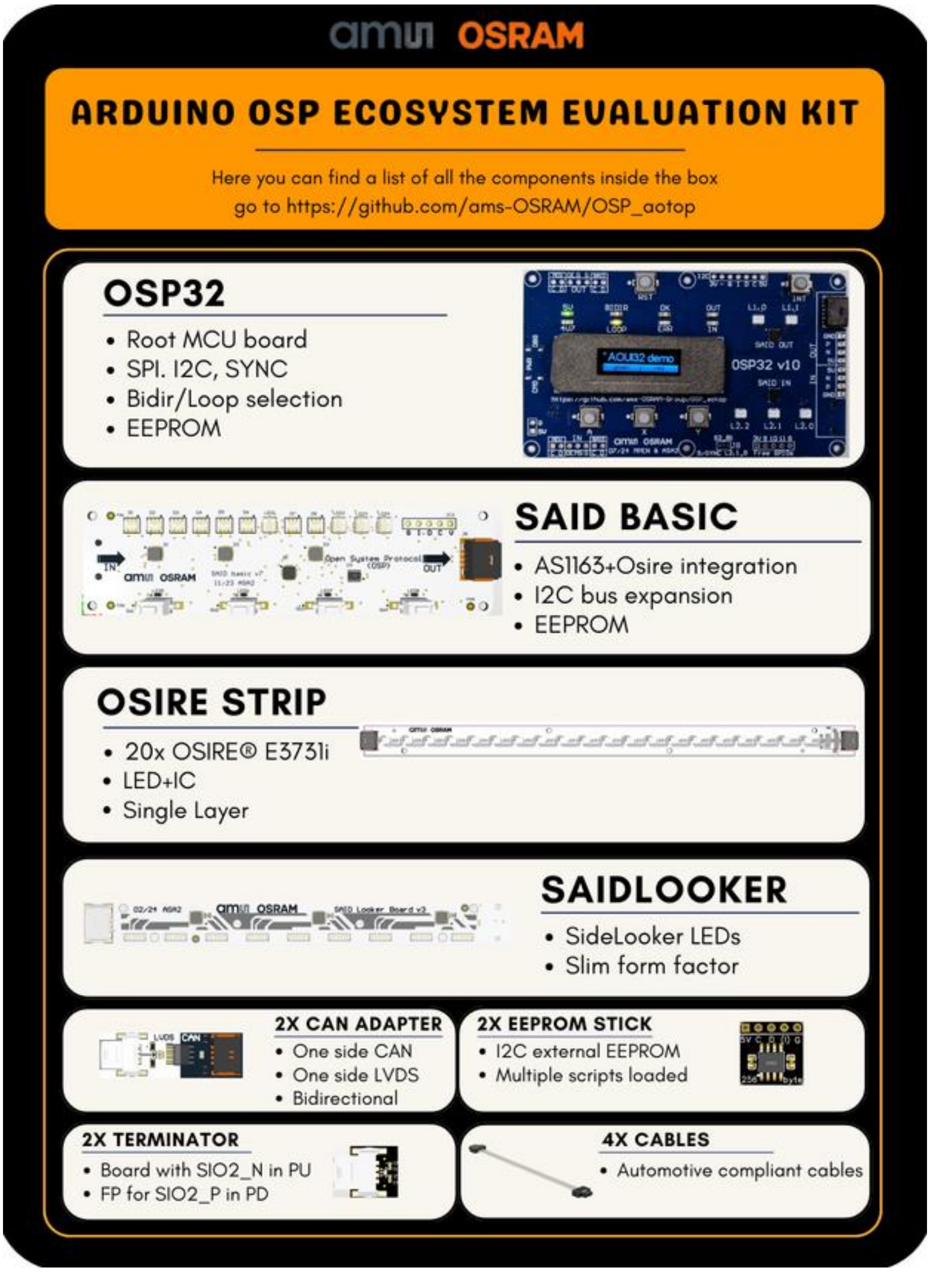
Thanks to its special features, AS1163 can control up to 9 non-multiplexed light sources thanks to its high resolution PWM engine. It is possible to use 3 channels as an I²C master bus. This gives the system great versatility, letting connect with the same device both light sources and sensors.

The E373i offers an integrated solution of the light emitter and its driver inside a compact package. One of the most important features then is the opportunity to design boards with a very small form factor.

2 What's in the box

The following table lists the board included in the ams OSRAM Arduino OSP ecosystem evaluation kit.

Figure 5: Arduino OSP ecosystem evaluation kit



The hardware on each board is detailed in separate documents.

Please refer to these guides for the schematics of the hardware:

- OSP32
- SAID basic
- Osire strip
- SAID Looker
- CAN adapter
- EEPROM stick
- Terminator

3 Let's play

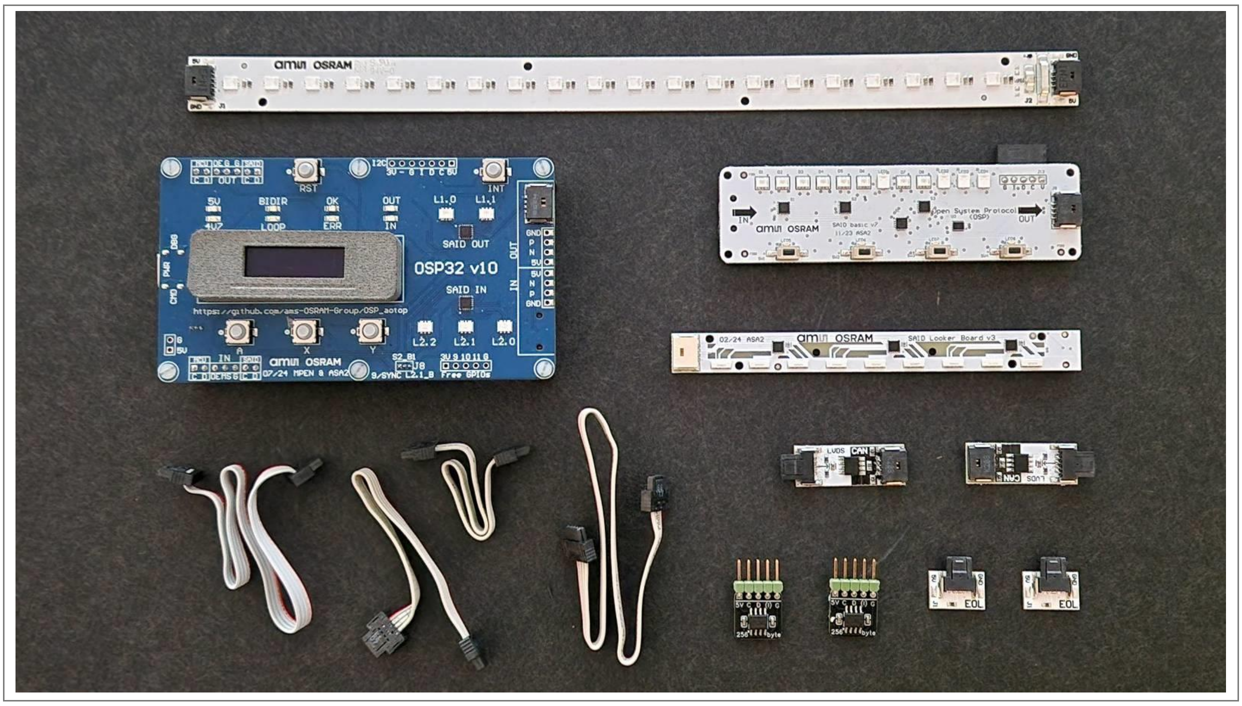
The EVK has already preprogrammed (the MCU is flashed with a demo). It's sufficient to power the boards up to see animations. It is useful to connect a PC to the USB port labeled CMD for advanced interactions.

If the user would like to expand or experiment additional ingredients are needed: Arduino IDE, board package (compiler) and OSP libraries. Every step necessary to control the system through the laptop is deeply described in the getting started section of the software libraries, in chapter 4, where you can also reflash the pre-programmed demo after your own experiments.

3.1 Familiarizing with the boards

It is possible to connect the boards in multiple ways. The ams OSRAM evaluation kit offers a whole portfolio to discover each of the features included in the devices shown in Figure 6 below.

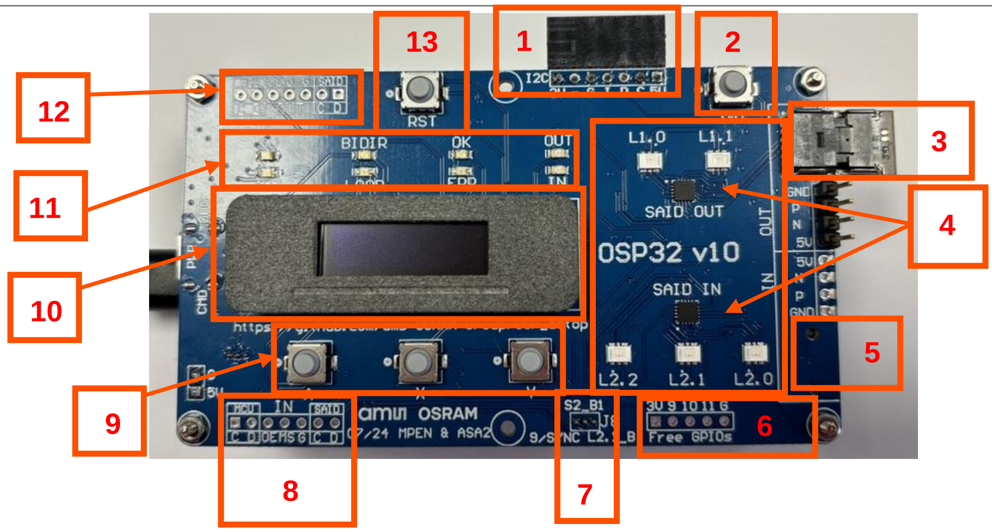
Figure 6: EVK content



The first board to be used (plugged into the PC) is the OSP32 board, the root MCU. This board powers and gives logic to the overall system.

3.2 Root MCU - OSP32 board

Figure 7: OSP32 board detailed



1 I²C header
2 Interrupt pin
3 Out connector
4 AS1163 + LEDs
5 IN connector
6 Free GPIOs
7 Configurable HW SYNC
8 Debug header IN
9 UI buttons
10 OLED
11 Signaling LEDs
12 Debug header OUT
13 Reset button

This board can run stand-alone.

The board can be powered from multiple points, to allow flexibility and unique features. Every USB port is labeled. The USB-C PWR port allows the connection with pure 5V. This is different from the USB-C ports DBG and CMD which are from the ESP32. They are in fact limited by the protection diodes and offer less than 5V (4.7V). These are used instead to interact with the laptop. CMD can be used as UART interface (flashing firmware and command line interaction) while DBG is meant to use the board in debug mode. Another power feeding can be done with the low left corner 2-way pin strip header labeled with 5V and G.

Starting from the upper left corner of the board, it is possible to analyze using a logic analyzer or an oscilloscope the signal going out from the ESP32 to control the first AS1163.

Symmetric to this pin header, there is the corresponding one for the response telegrams from the daisy-chain to the MCU. This gives the opportunity to complete monitoring the signals and the messages transferred. There are two others pin headers exposed: one is on the top edge of the board, a pin header with the I²C bus of SAID OUT exposed. On the bottom side instead, the ESP GPIOs not used are exposed for future usage and programming.

There are also two buttons on the top part of the board. One to reset the ESP, the other to send an interrupt to the I²C controller in SAID OUT. They can be considered parts of the overall user interface (UI). This includes the OLED display in the center and the A, X and Y buttons below. Button functions so far deployed allow modifying the system settings, in demo apps, for example the buttons change the dimming level of the output or the timing of the animation.

Above the OLED there are two rows of signaling LED with the respective label to let the user know some valuable information like the configuration or the error state.

To the right of the OLED we find, vertically aligned, two AS1163. The upper one controls just two LEDs; the third channel is configured as I²C. It has been exposed to be connected to an embedded EEPROM and the bus has been exposed in the I²C strip header above it too. This device is wired for MCU mode (and configured for flavor SPI). The MCU in this case is controlling the system with a standard 2-wire SPI connection. The lower SAID controls three LEDs but optionally channel 1 can be used as hardware SYNC pin.

3.3 Wiring OSP chains

Figure 8: OSP chain configurations

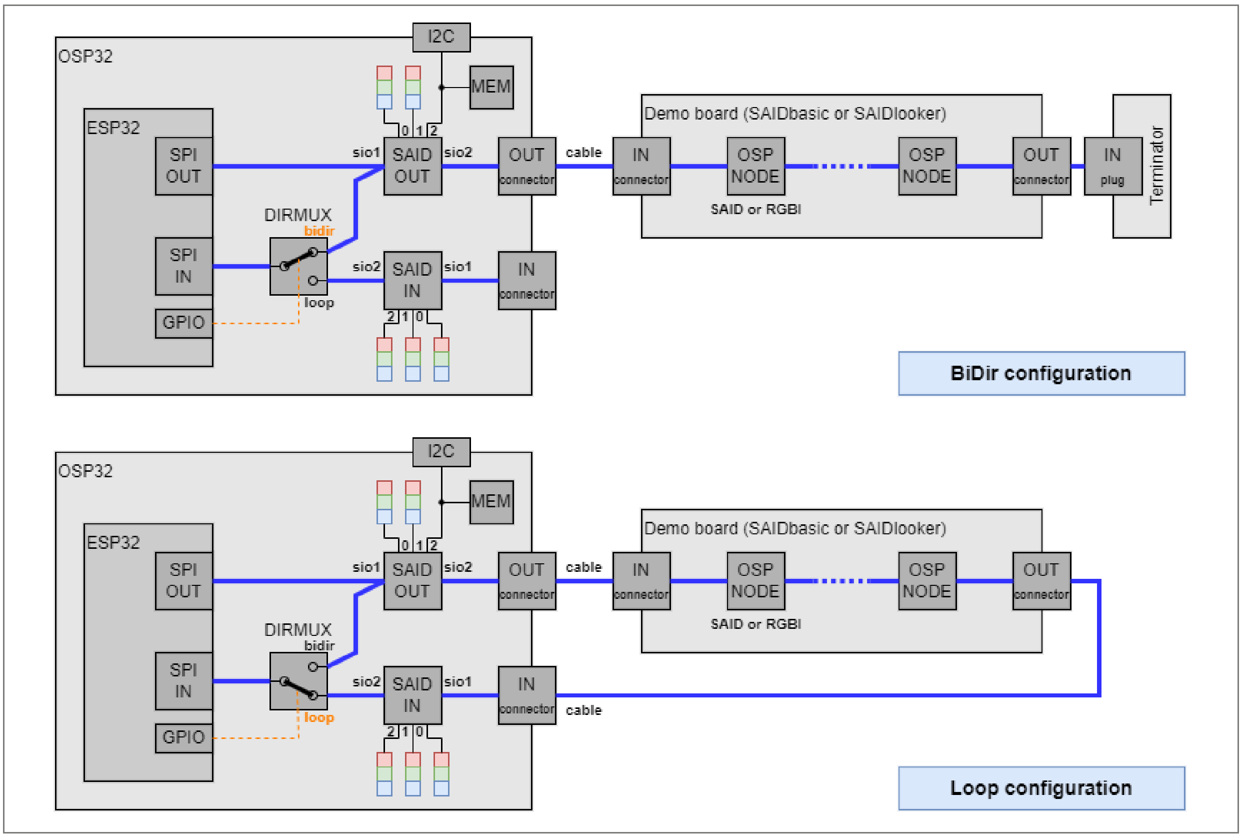


Figure 8 shows the two possible connections in a typical application.

It is mandatory to always connect a terminator at the end of the chain or use a cable to connect the end of the chain back to the IN port of the root MCU board.

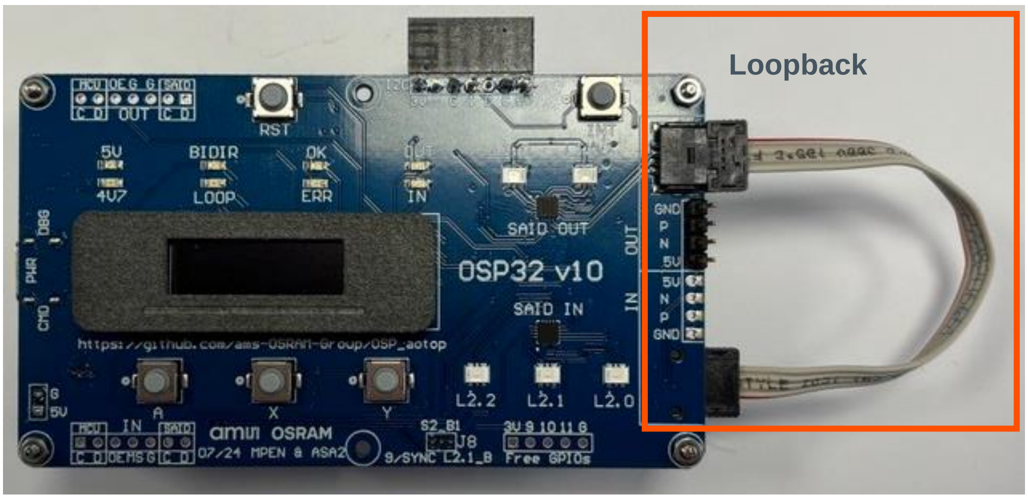
You can either test the bidirectional configuration by just placing the terminator resistor to the connector labeled OUT, as shown in Figure 9 below.

This is the shortest OSP chain: MCU - SAID OUT – terminator.

Figure 9: Bidir configuration

It is possible to evaluate the loopback mode with just this board too. It is sufficient to connect the cable between the OUT and IN connectors. This closes the loopback directly into the SAID IN device on the OSP32 board as displayed in the next picture Figure 10. This creates a two nodes chain: MCU – SAID OUT – SAID IN – MCU.

Figure 10: Loopback configuration



Now, by running one of the examples of the SW library (chapter 4), you can interact with this board. By running the software included with the EVK, you can exploit the various OSP features.

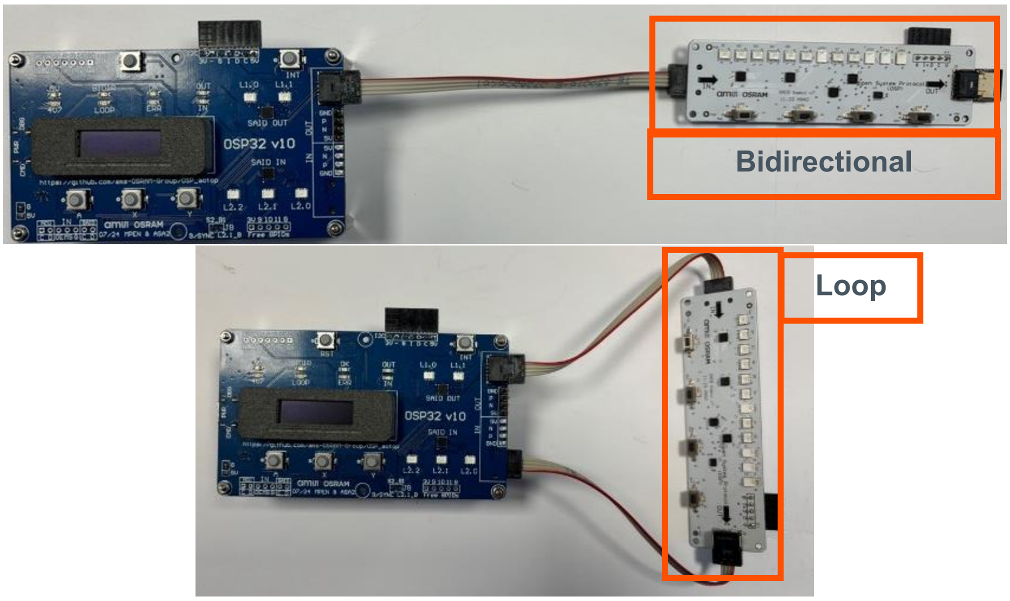
To summarize, this is the list of the main features included in this board:

- SPI interface between the MCU and SAID OUT
- I²C bus connected to an EEPROM and exposed for external boards
- Input mux controlled by MCU allowing detection of the INIT configuration
- Hardware SYNC
- UI to control and interact with apps
- Plug & Play
- Multiple power feed options
- Trace headers
- Exposed free GPIOs for future usage

3.4 Plug the daughter boards

After assessing the working principles of the main board, it's time to connect the daughter boards too.

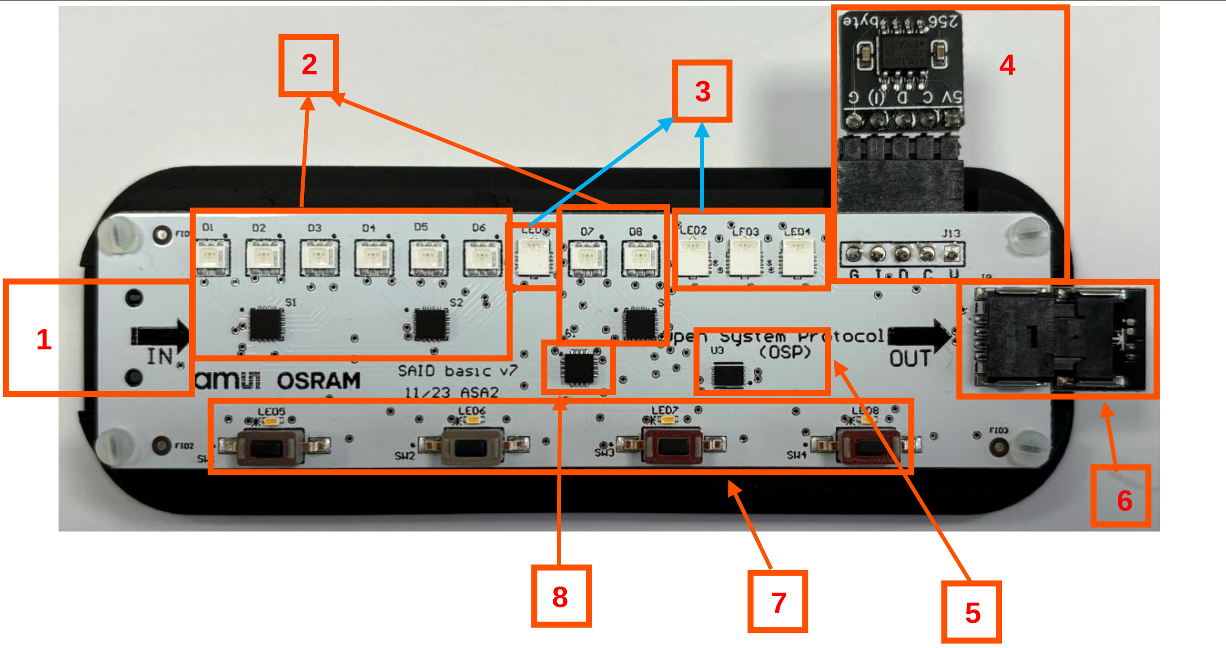
Figure 11: OSP32 + SAID basic board both in bidirectional and loopback mode



The first board to be plugged is the SAID basic board.

3.4.1 SAID basic board

Figure 12: SAID basic board detail



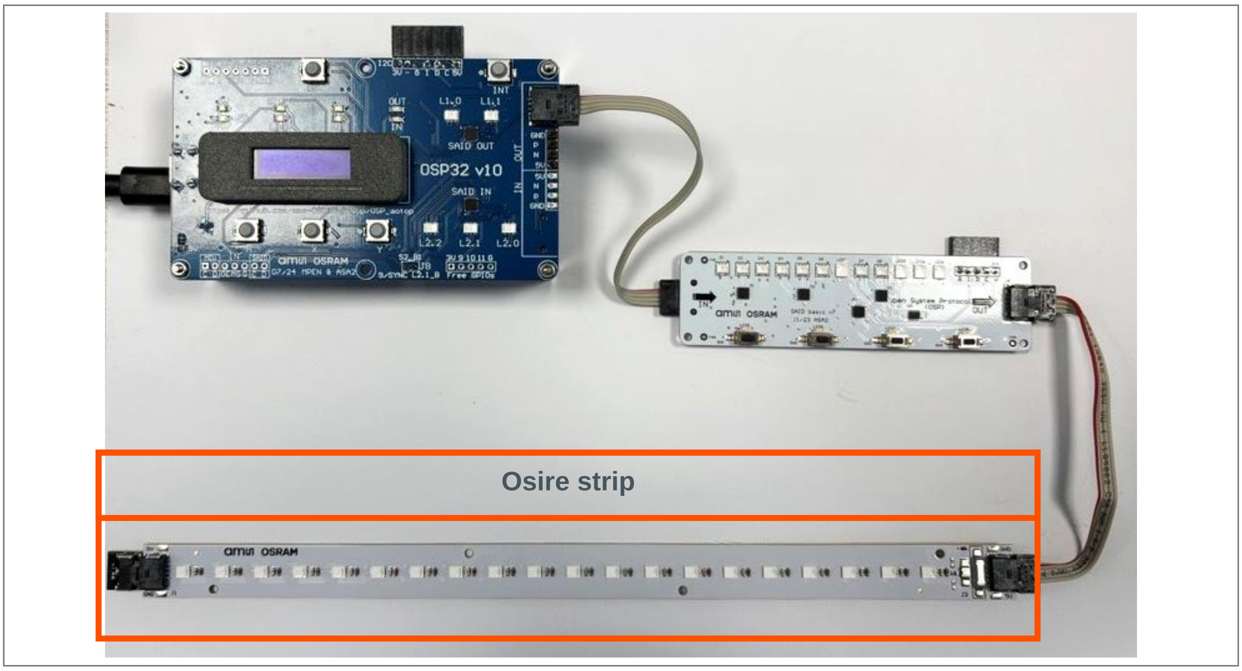
1 IN connector (bottom)
2 AS1163 + LEDs
3 Osire E373i
4 I²C header +external EEPROM
5 Internal EEPROM
6 OUT connector + terminator
7 UI buttons + indicator LEDs
8 I/O-expander

The SAID basic board includes both AS1163 and OSRAM OSIRE® E3731i together and daisy chained. Intentionally, it has been designed to mix these OSP devices to show the OSP interoperability of different devices.

With this board it is possible to explore the I²C bus. The last AS1163 is connected to an I/O-Expander to control 4 buttons and 4 white LED indicators. In addition, the I²C bus connects to an EEPROM where scripts/calibration data can be stored. The factory default image of this EEPROM is an animation script for a demo. So EEPROM experiments are better performed with the EEPROM on the OSP32 board. If the EEPROM on the SAID basic board gets modified, rest assured, there is a sketch in the OSP library to reflash the EEPROM.

3.4.2 Osire strip board

Figure 13: OSP32 + SAID basic + Osire strip + terminator



This board has been designed to show the LEDs OSIRE® E3731i. The device is an intelligent RGB device, incorporating besides R/G/B LEDs some embedded logic that implements the OSP protocol and the drivers for the three R/G/B LEDs. It is specifically designed for automotive interior applications with high dynamic RGB lighting scenarios.

Figure 14: Osire strip detail

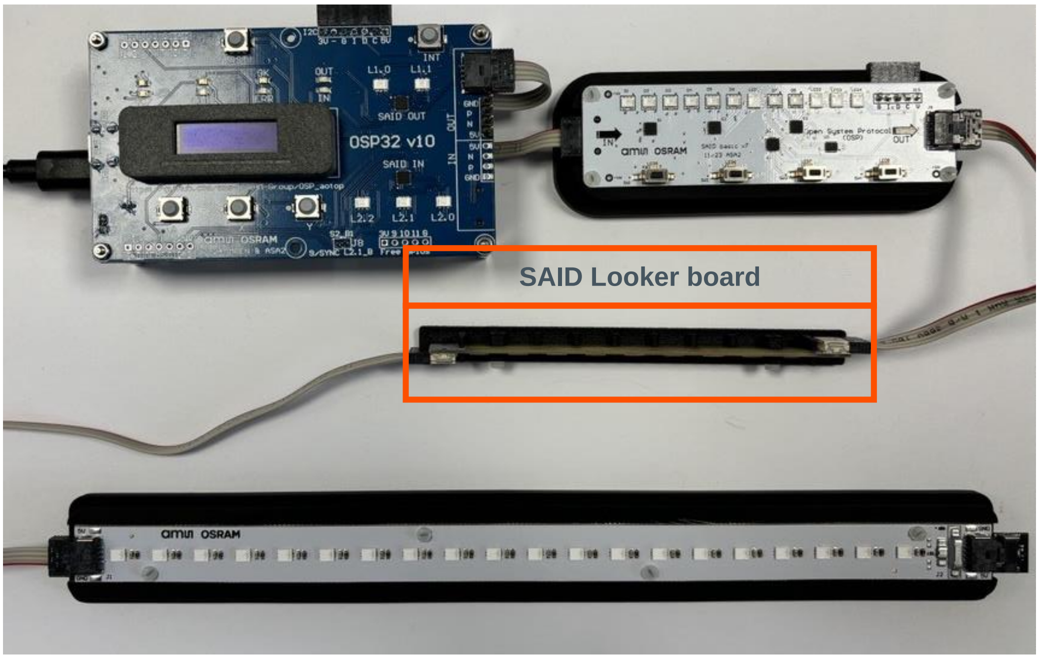
1 Osire E373i
2 OUT connector + terminator

PCBs for E373i devices can be single layer design. The device developed can be integrated into a system where small and compact designs are mandatory.

3.4.3 SAID looker board

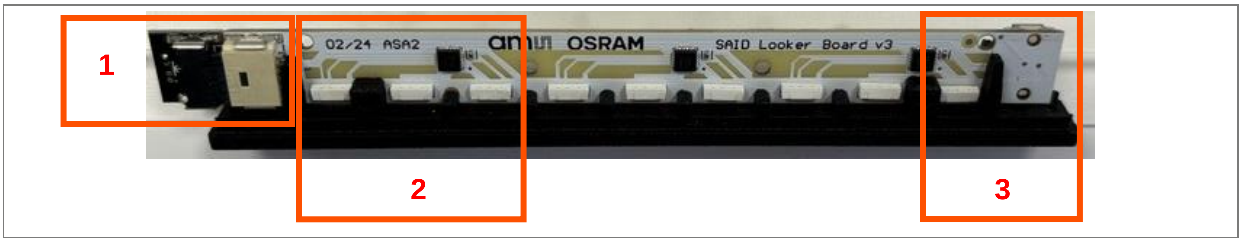
The board in Figure 15 has been included in the EVK to show another component of the light emitters portfolio for interior lighting, namely side lookers. This board has three SAIDs, each driving three RGB side lookers (E5515).

Figure 15: OSP32 + SAID basic + SAID looker + Osire strip + terminator



The combination of AS1163 with E5515 allows potentially to reach sections in the car with a flat PCB design and without the necessity of coupling the light with special lightguides. AS1163 can control 3 different RGB E5515. It offers then the flexibility of a MCU less configuration along with the opportunity to place the board in places where "normal" top emitters would require a special design.

Figure 16: SAID looker board detail



1 OUT connector + terminator
2 AS1163 + 3×E5515
3 IN connector (bottom)

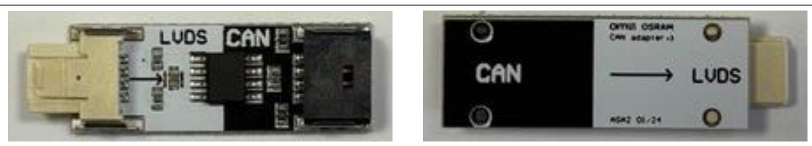
Information:

Please note that the IN connector is on the right-hand side when the texts are readable. OSP ports are rather symmetrical, so no harm occurs when the IN and OUT port are swapped, but the animation effects might not be what you expect.

3.4.4 CAN adapter

To show the automotive compliance of our devices, it is possible to place a CAN adapter board on both sides of an OSP cable.

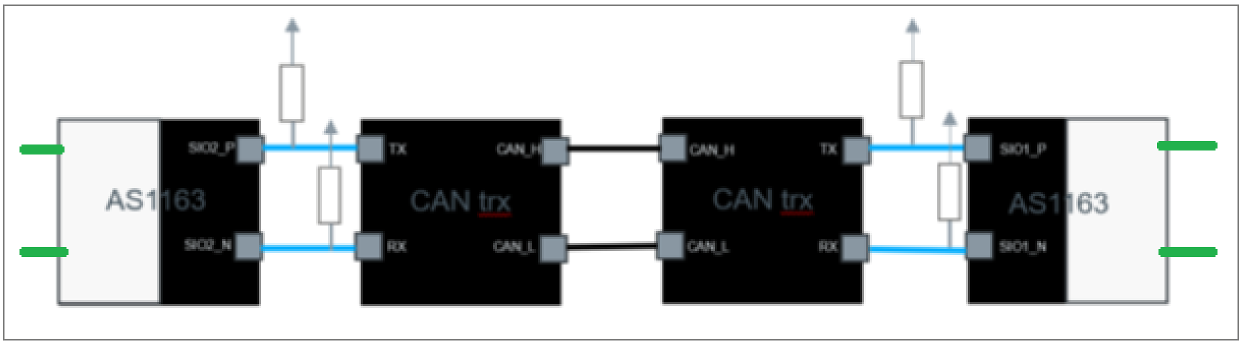
Figure 17: CAN adapters



(1) CAN adapters: Top & bottom layer

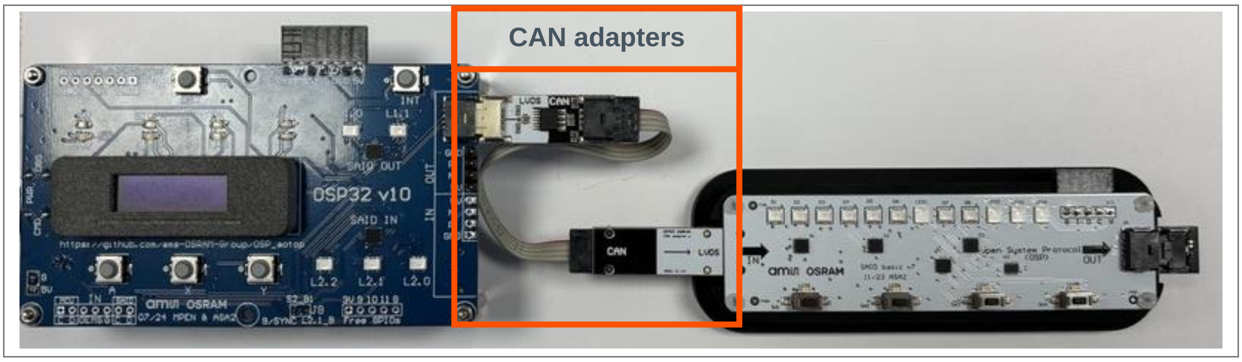
No further programming or SW edits are necessary. The devices autodetect the different physical layers and are fully compatible with the CAN translation.

Figure 18: Example of CAN PU/PD network



Two boards are necessary to translate the LVDS into CAN domain and the other way around for the next device.

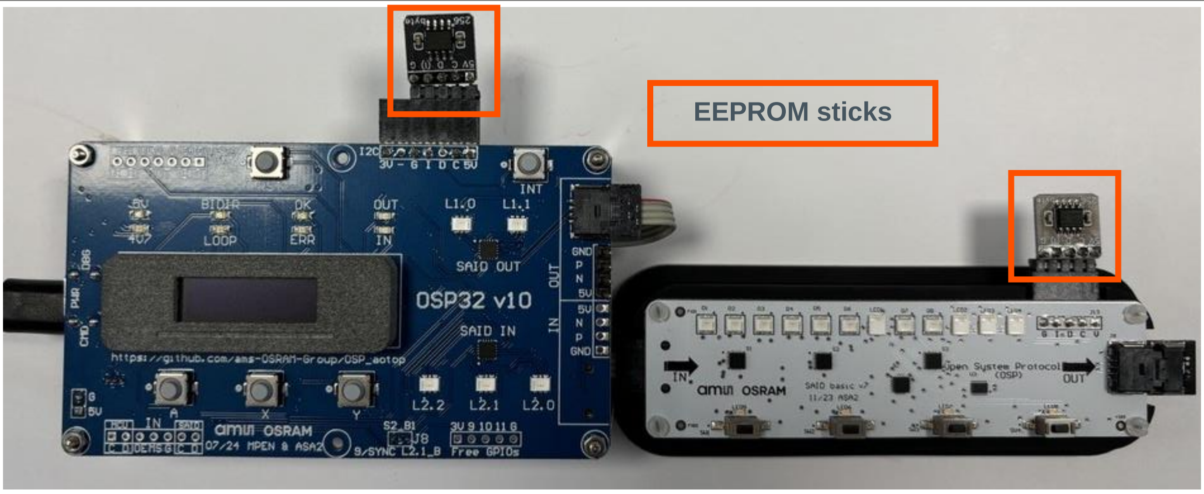
Figure 19: OSP32 + CAN adapters + SAID basic



3.4.5 EEPROM stick

It is possible to connect external EEPROMs too, provided in the EVK. Both the OSP32 and the SAIDbasic board have a SAID which is configured as I²C bridge and which have that I²C bus pinned out on a header.

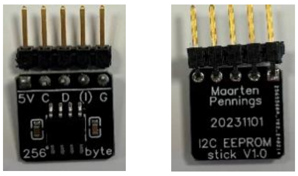
Figure 20: OSP32 + EEPROM sticks + SAID basic



A plugged-in EEPROM stick can be accessed (via the SAID) by the ESP32. This is used in demo software: sketch saidbasic.ino loads an animation script from an external EEPROM stick, when available.

There is a sketch (eepromflasher.ino) to flash the EEPROM with your own animation script. Please be aware that the EVK comes with two EEPROM that have been flashed with two different animation scripts (bouncingblock and colormix). The EEPROMs are not write protected. If they are modified eepromflasher.ino allows you to restore the original content.

Figure 21: EEPROM stick



(1) EEPROM stick: Top & bottom layer

Please notice that the OSP32 and the SAID basic board have different I²C header. The OSP32 is a 7-way socket to include the 3V3 domain too. It is separated from the "classic" 5-way socket thanks to an empty pin. The basic board instead exposes the shorter 5-way socket (5V, GND, SCL, SDA, and INT). The labeling is the same except the additional pins on the OSP32.

3.4.6 Terminator

All the OSP devices need to be addressed. This ensures the correct delivery of commands. Each node has a configurable address. This is established during the start-up phase with the initbidir or initloop command. To determine the total number of nodes, the chain must have a special end of line (EOL) configuration. This configuration is based on a pull up/pull down network. In particular, the SIO2_P line of the last device must be in PD (pull-down) configuration while the SIO2_N line must be in PU (pull-up) configuration.

To keep the modular approach, the following board has been designed and included in the EVK.

Figure 22: Terminator board

Thanks to the weak internal pull down of the AS1163 and of the Osire E373i, it is sufficient to place just the SIO2_N in pull-up configuration.

4 Software

The hardware described in the last chapters cannot work without the right software.

The OSP protocol in particular has its telegrams that need to be generated and forwarded to the devices. This is possible thanks to the integration of the ESP32 MCU with OSP libraries.

The *Arduino OSP evaluation kit* comes with software; the *aolibs*, short for Arduino OSP libraries from ams OSRAM. Figure 23 shows the libraries and their dependencies.

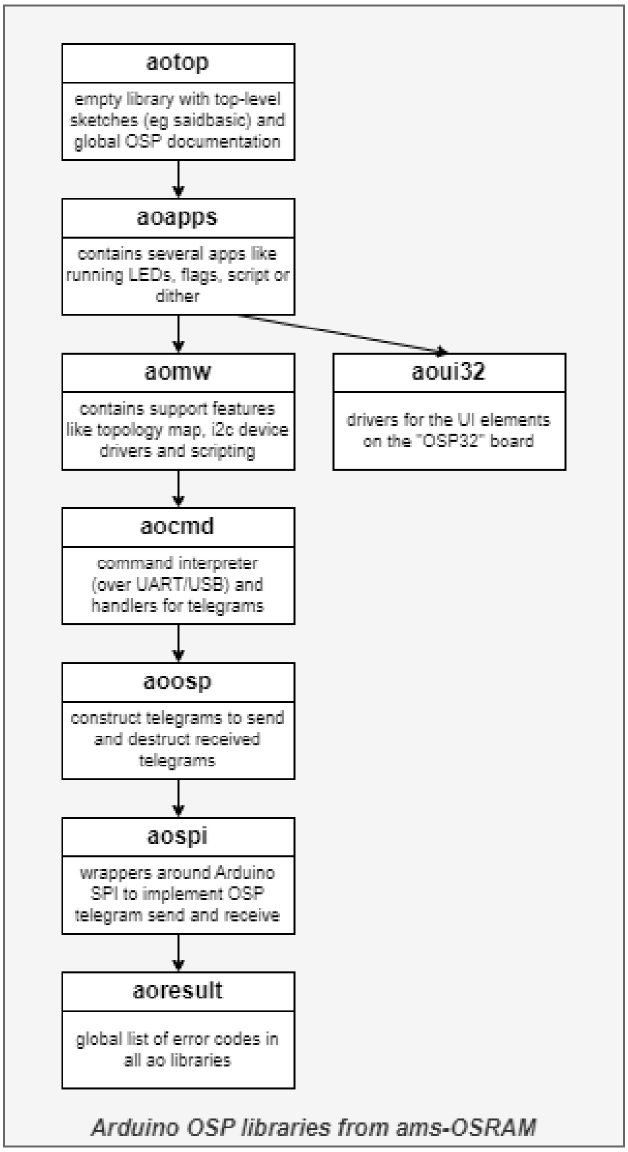
ams OSRAM has an organization page on GitHub (github.com/ams-OSRAM)

The *aolibs* are publicly available on the ams OSRAM GitHub page; the landing page is github.com/ams-OSRAM/OSP_aotop

It links to various resources, one of them is the Getting Started manual github.com/ams-OSRAM/OSP_aotop/blob/main/gettingstarted.md

The Getting Started explains how to install the Arduino IDE, how to add the ESP32 boards (compiler) to the IDE, how to add the *aolibs* to the IDE, and finally it shows how to run an example on the Arduino OSP evaluation kit.

Figure 23: Arduino OSP libraries structure



5 Appendix

5.1 List of abbreviations and acronyms

Table 1: List of abbreviations and acronyms

Abbreviation	Definition
OSP	Open system protocol
EVK	Evaluation kit
SAID	Stand-alone intelligent driver
LED	Light emitting diode
RGB	Red, green blue
PWM	Pulse-width modulation
I²C	Inter-integrated circuit
SPI	Serial peripheral interface
EEPROM	Electrically erasable programmable read-only memory
MCU	Microcontroller unit device; also OSP device’s port configuration
OUT	Output
IN	Input
SW	Software
HW	Hardware
USB-C	Universal serial bus with C style connector
PWR	Power
DBG	Debug
CMD	Command
UART	Universal asynchronous receiver transmitter
UI	User interface
GPIO	General purpose input output
I/O-expander	Input/Output expander (I²C device)
PCB	Printed circuit board
CAN	Controller area network
LVDS	Low-voltage differential signaling
CAN trx	CAN transceiver
3V3	3.3 volts
5V	5 volts

Abbreviation	Definition
EOL	End of line – OSP device’s port configuration
SIOx_P	Serial input output (port x) Positive
SIOx_N	Serial input output (port x) Negative
PU	Pull up
PD	Pull down

6 Revision information

Definitions

Draft / Preliminary:
The draft / preliminary status of a document indicates that the content is still under internal review and subject to change without notice. ams-OSRAM AG does not give any warranties as to the accuracy or completeness of information included in a draft / preliminary version of a document and shall have no liability for the consequences of use of such information.

Changes from previous version to current revision v1-00	Page
Initial production version	

- Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
- Correction of typographical errors is not explicitly mentioned.

7 Legal information

Headquarters

ams-OSRAM AG

Tobelbader Strasse 30

8141 Premstaetten

Austria, Europe

Tel: +43 (0) 3136 500 0

Please visit our website at ams-osram.com

For information about our products go to Products

For technical support use our Technical Support Form

For feedback about this document use Document Feedback

For sales offices and branches go to Sales Offices / Branches

For distributors and sales representatives go to Channel Partners