Sense the power of light

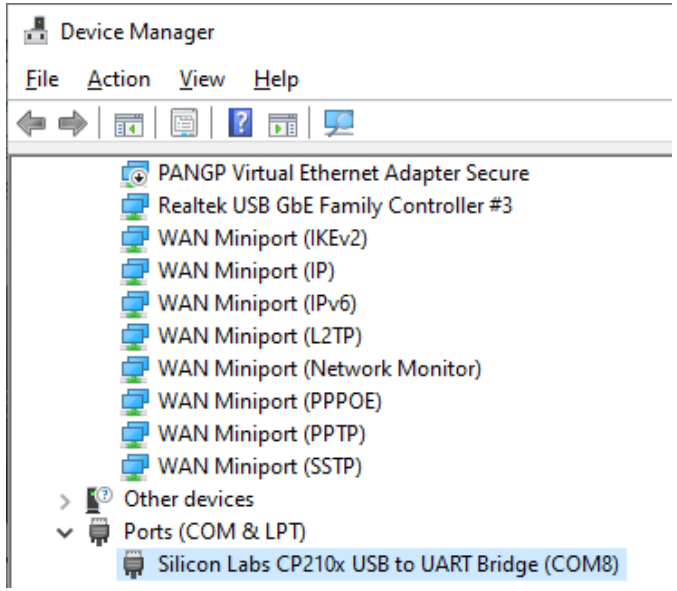
AS1170 EVK GUIDE

Hardware Description and SW examples

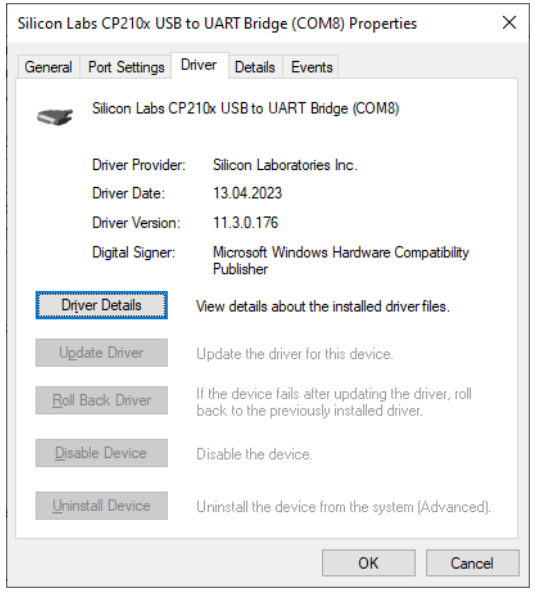
GUI and Driver Installation

step by step guideline

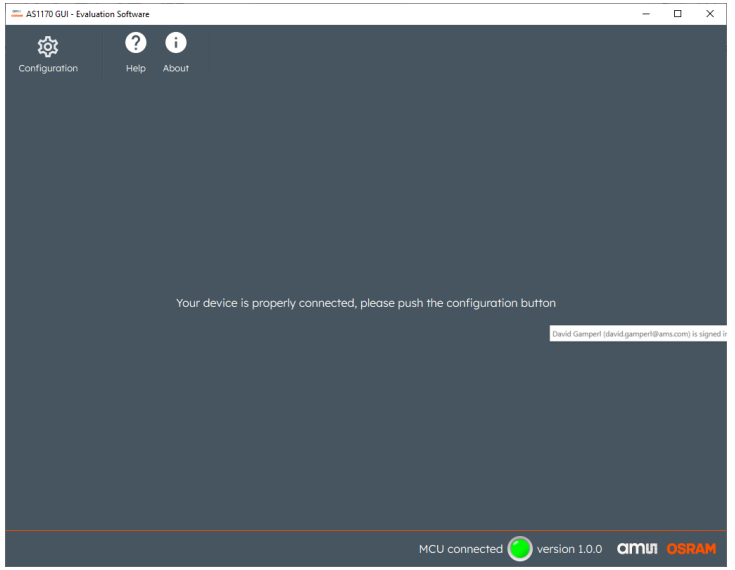
1) Unzip/Extract SW package to local hard drive
2) Connect Board via µUSB cable to AS1170 EVK
3) Open Device Manager and search for “USB to UART Bridge”
4) Select “Update Driver” from your local drive and select folder of step #1 (location of extracted SW package)
5) After Driver is successfully installed launch AS1170_GUI.exe
6) Correct installation will show green status lamp on bottom right of GUI SW



Step 3



Step 4

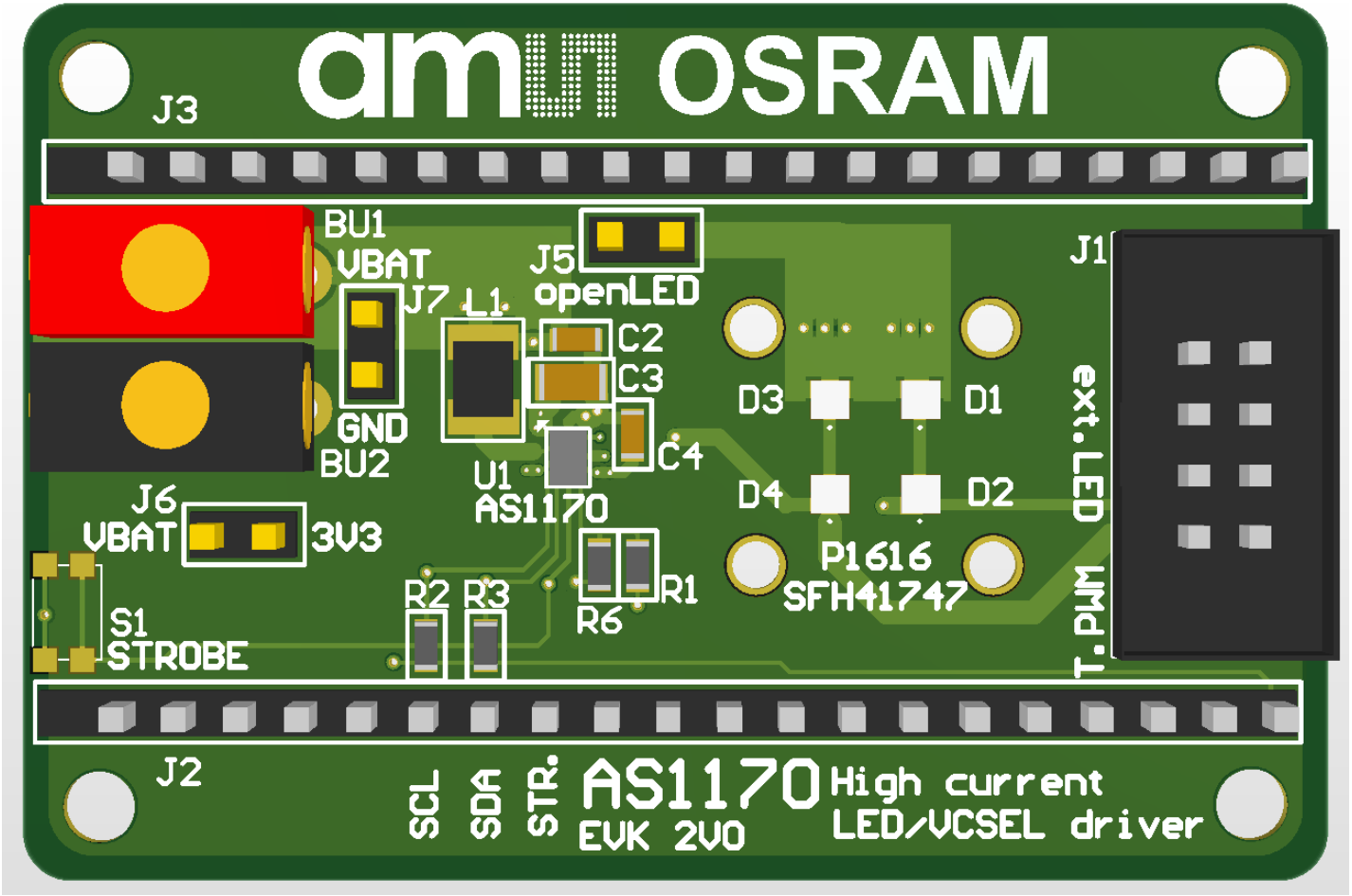


Step 6

AS1170 EVK Jumper and Terminal Description

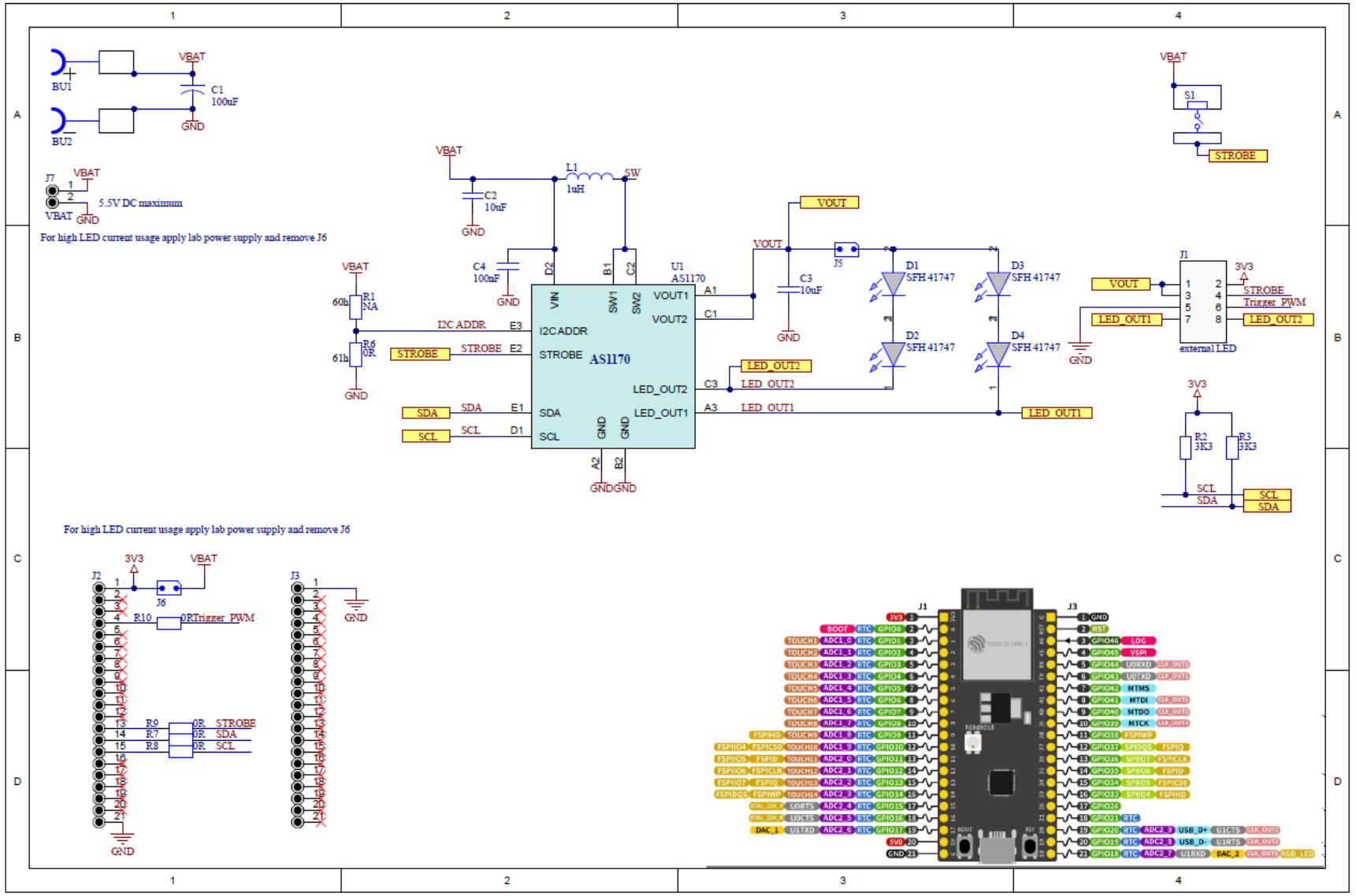
Layout and Jumper Description

Input	Description
BU1	**OPTIONAL: ONLY USE for high LED currents (>450mA). Use lab power supply and remove J6** VBAT 2.7V – 4.4V
BU2	GND
S1	Strobe button
J1	External LED Board connector When used remove J5 to disconnect on-board LEDs
J2	MCU Board Connector (SCL, SDA, Strobe, 3V3)
J3	MCU Board Connector
J5	Open/Disconnect onboard SFH41747 LEDs (remove when external LED Board is connected)
J6	VBAT to 3V3 jumper If set AS1170 is supplied by MCU Board with 3.3V – remove when BU1/BU2 are used.
J7	VBAT/GND measurement pin



AS1170 EVK Schematic

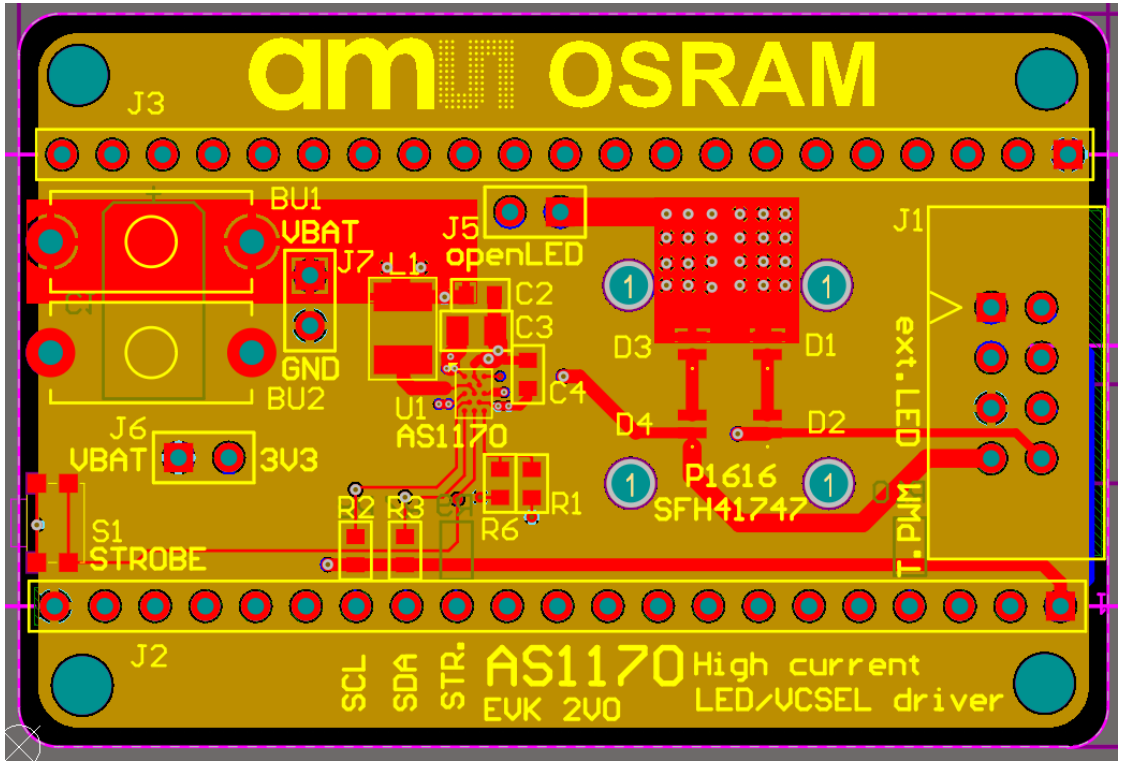
Circuit Diagram

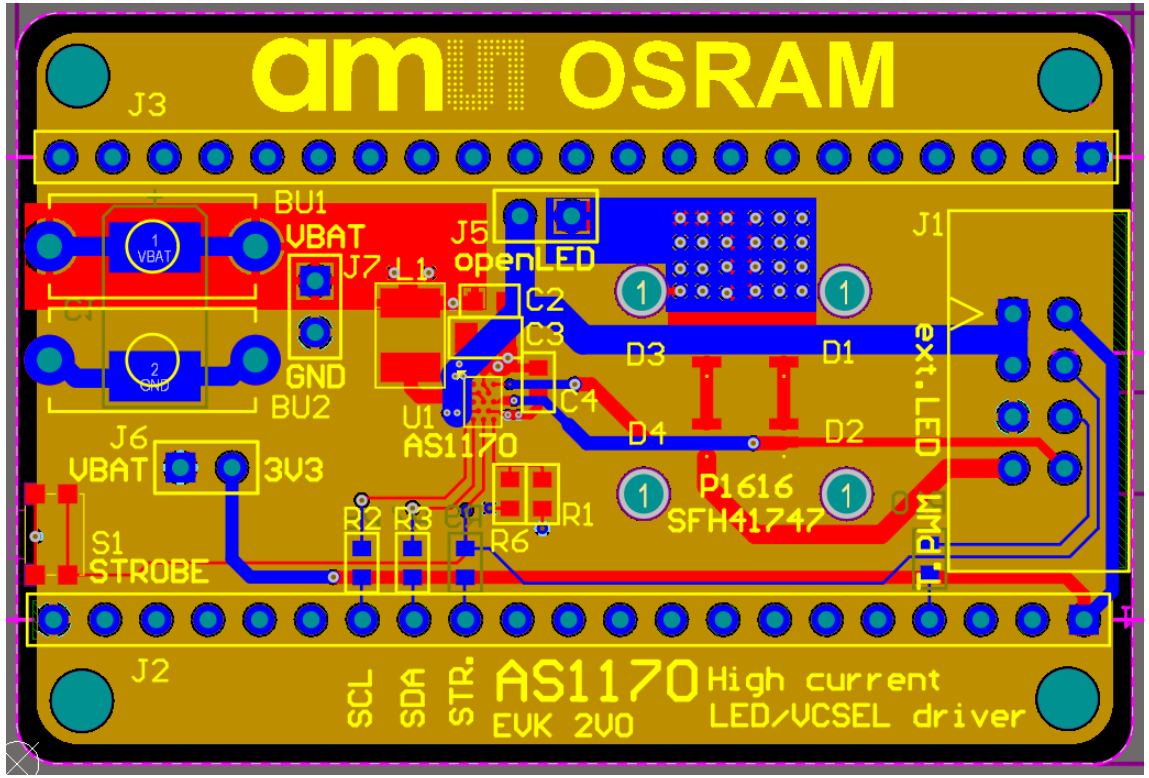


AS1170 EVK Layout (1)

All layers

Top: red
IN1: yellow (GND)
IN2: light blue (VBAT)
Bot: dark blue



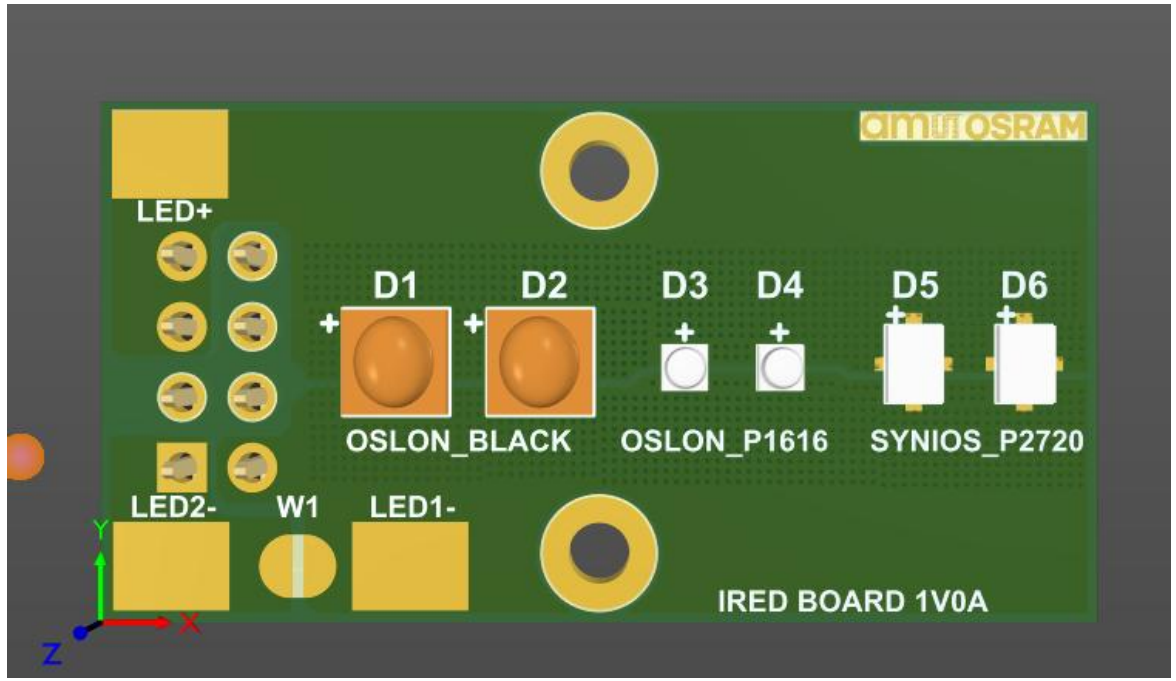


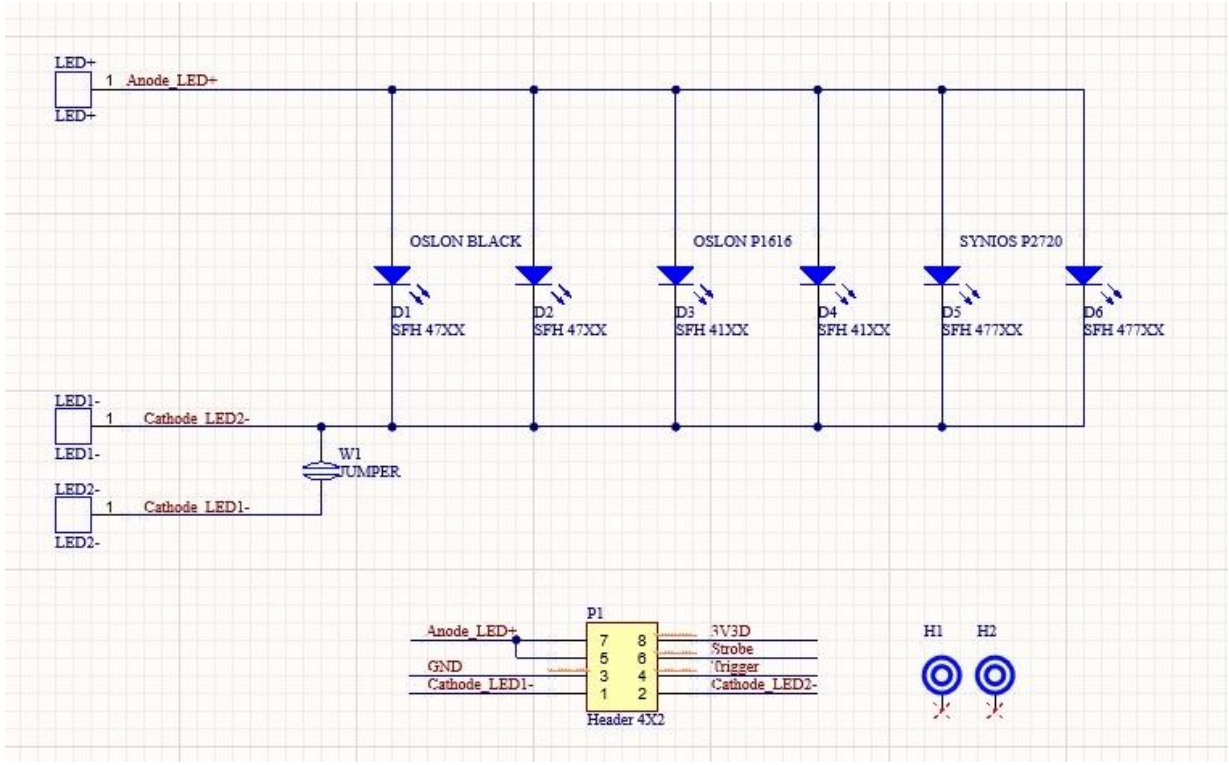
IR LED Break-out board schematic

Header P1 connected to J1 of AS1170 EVK

This break-out board can be used to assemble and test several different LED options.

Note: if break-out board is connected, make sure J5 of mainboard is removed to disable on-board LEDs

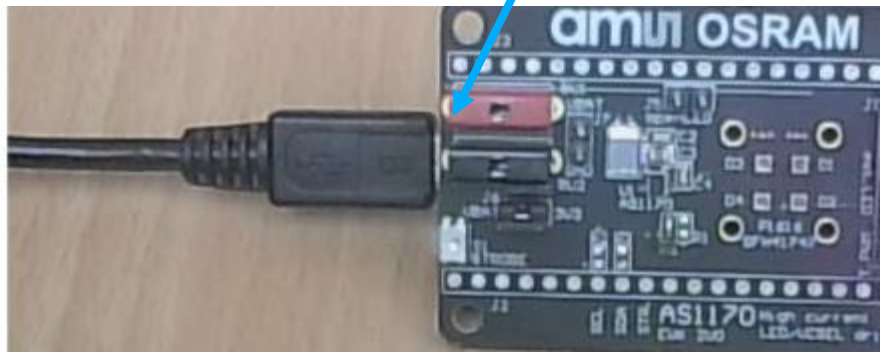


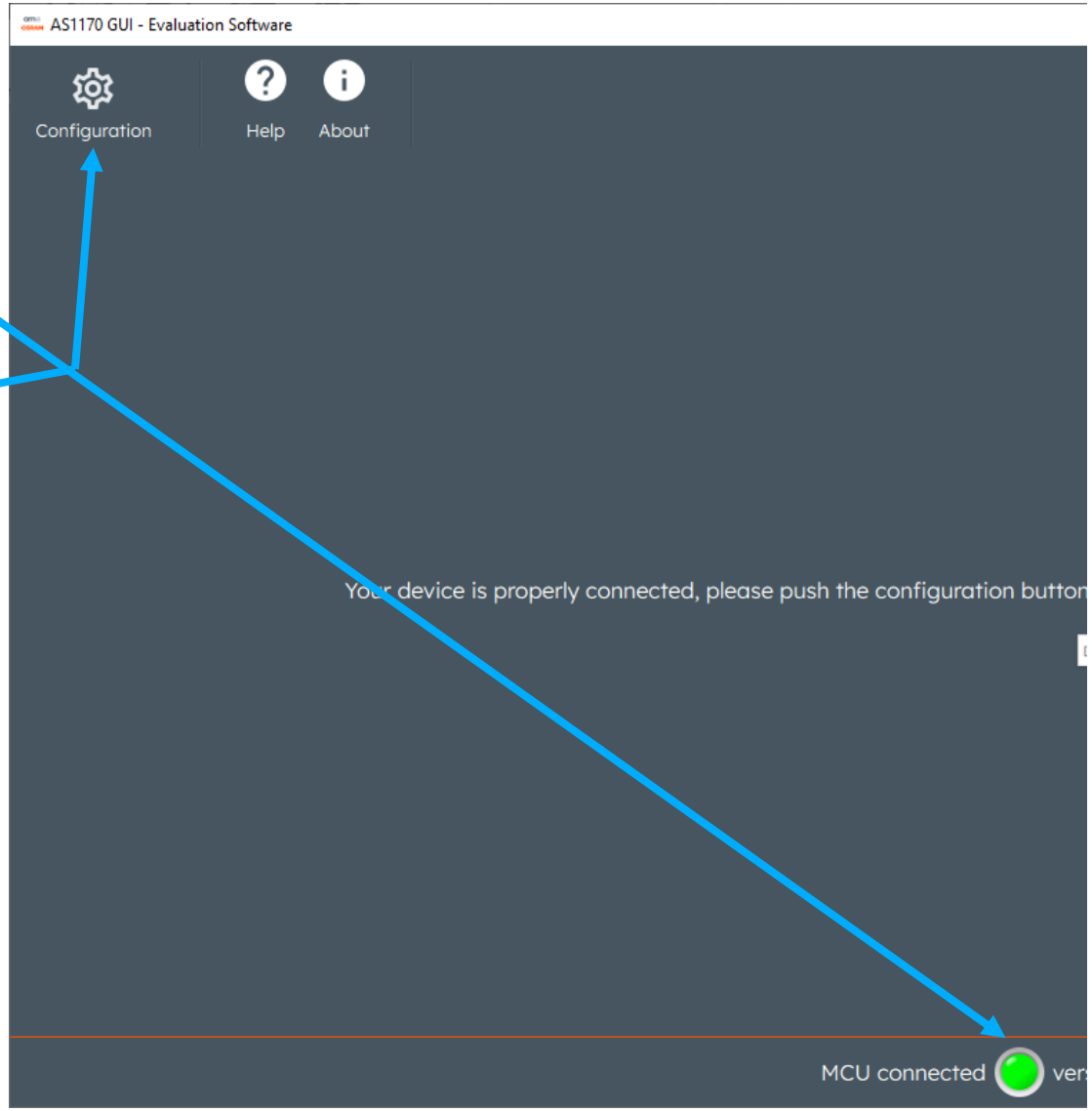


AS1170 GUI SW

First Steps

1) Connect EVK with µUSB Cable to USB Port of EVK
2) Start AS1170_GUI.exe
3) If USB_UART Driver is installed successfully green indicator will appear
4) Press “Configuration Button” to access device setup window

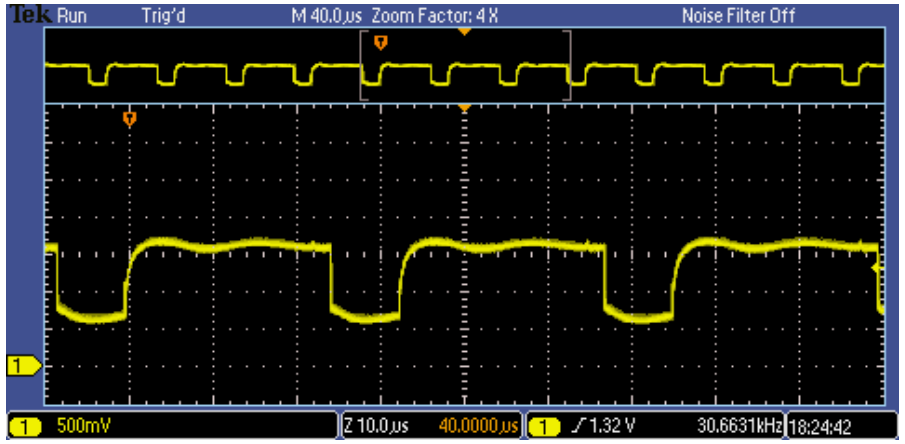




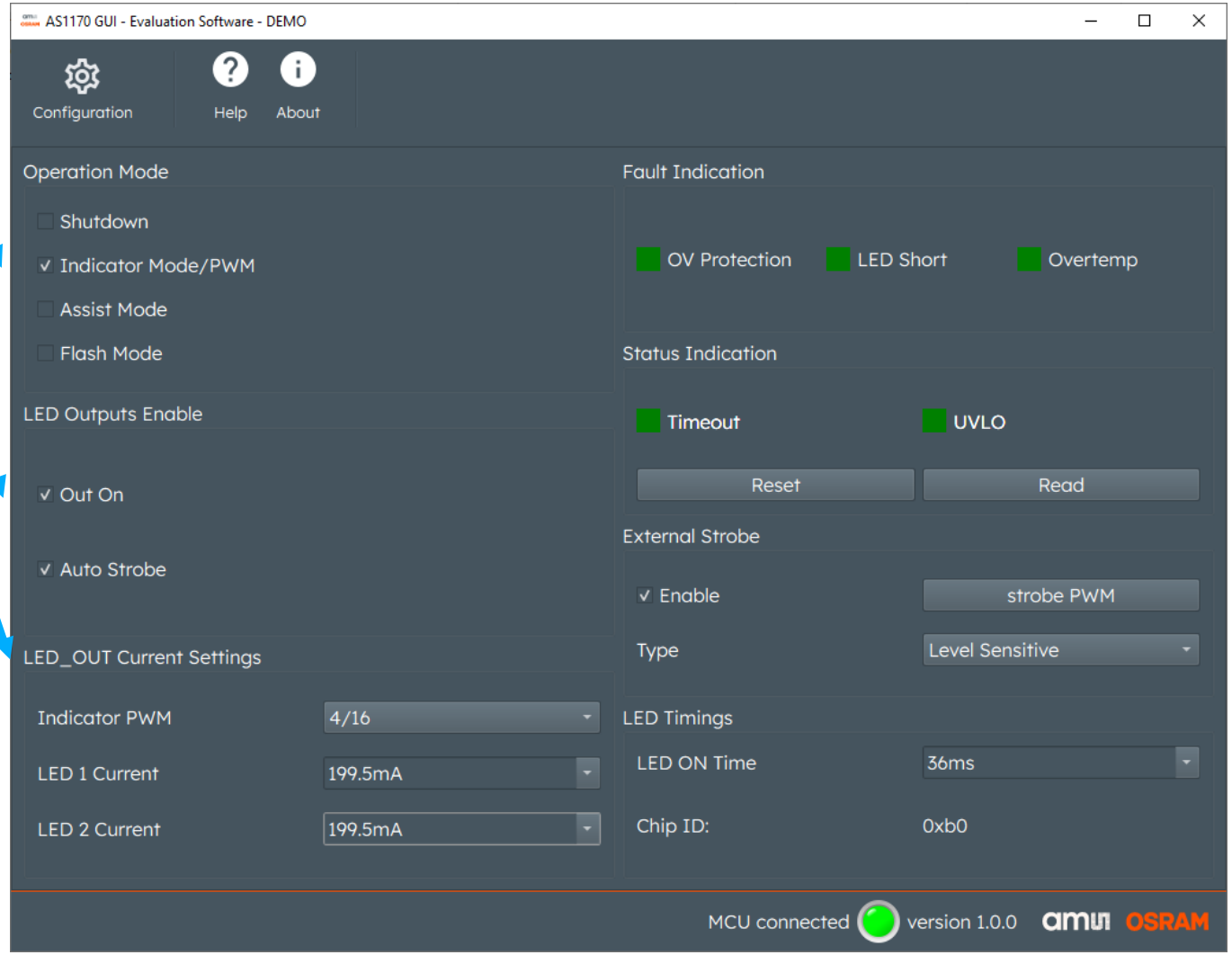
AS1170 GUI SW

How to configure “Indicator Mode/PWM”:

1) Enable “Indicator Mode/PWM”
2) Configure LED_OUT Current Setting
   - Indicator PWM 4/16 (=25% Duty Cycle)
   - LED 1 Current 199.5mA*
   - LED 2 Current 199.5mA*
3) Enable “Out On” Checkbox
   - LED enabled with ~400mA (LED_OUT1 and LED_OUT2 combined on break-out board
   - Duty Cycle 25% ; PWM_f 30.6 kHz



Yellow: LED_OUTx Waveform: 30.6kHz ; DC 25%

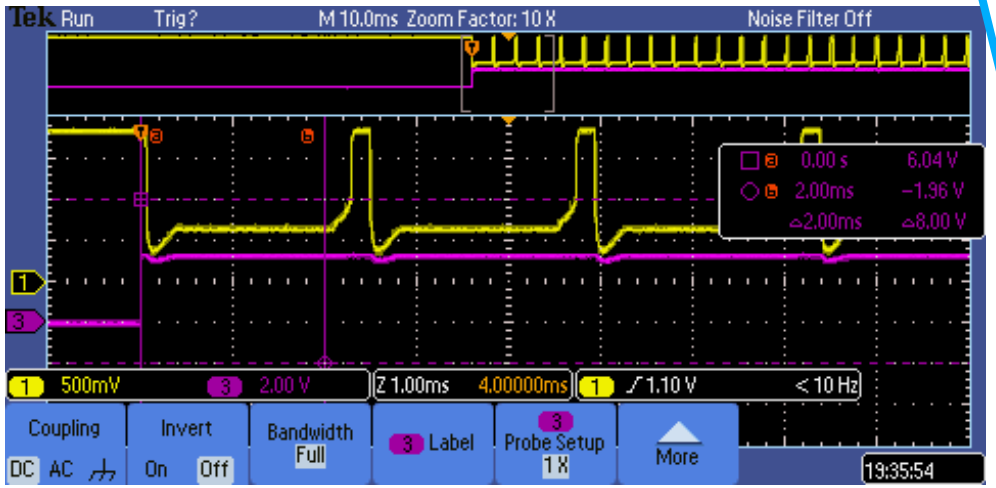


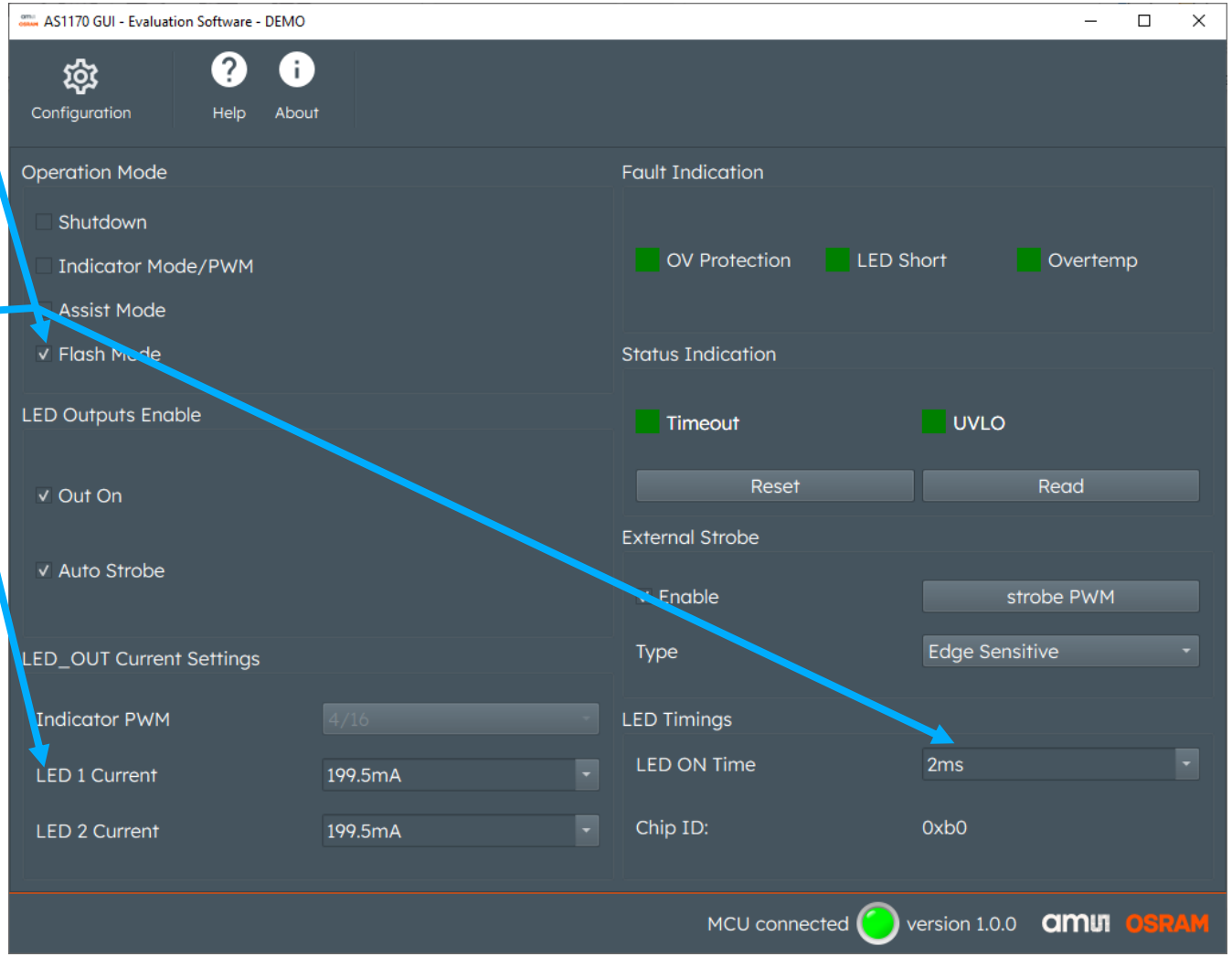
*Note: Don’t exceed 400mA total current via USB connection

AS1170 GUI SW

How to configure “Flash Mode”:

1) Enable “Flash Mode”
2) Configure LED_OUT Current Setting
   - LED 1 Current 199.5mA*
   - LED 2 Current 199.5mA*
3) Configure “LED Timings”
   - Select LED ON Time “2ms”
4) Enable “Out On” Checkbox
   - Press Button “S1” (purple waveform below)





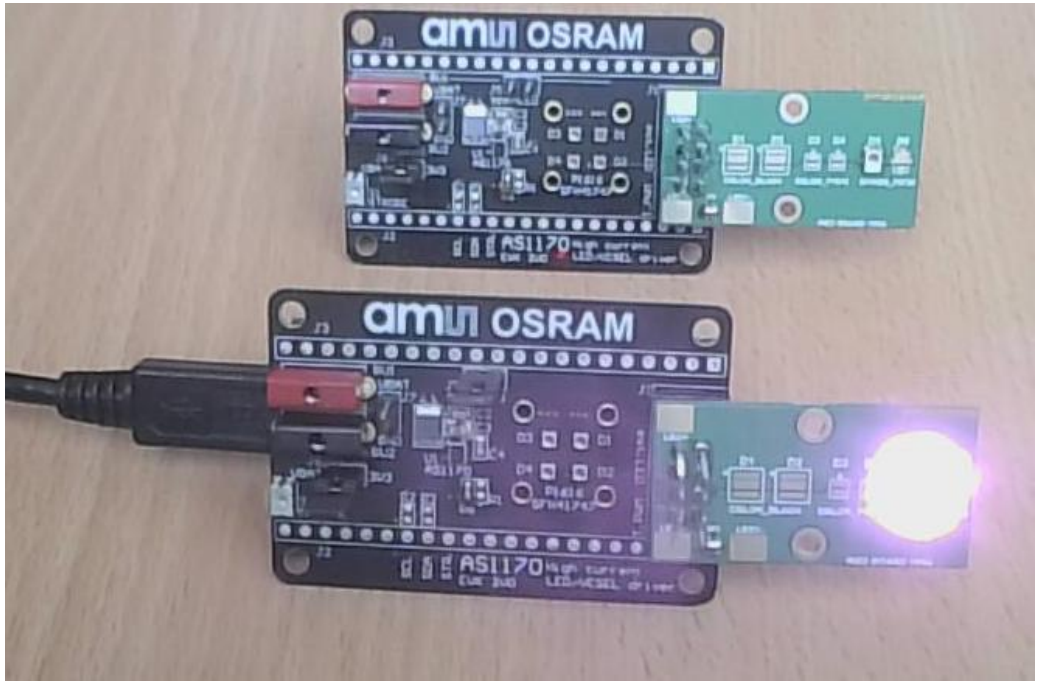
Yellow: LED_OUTx waveform with 2ms on time

Purple: Strobe signal via button S1

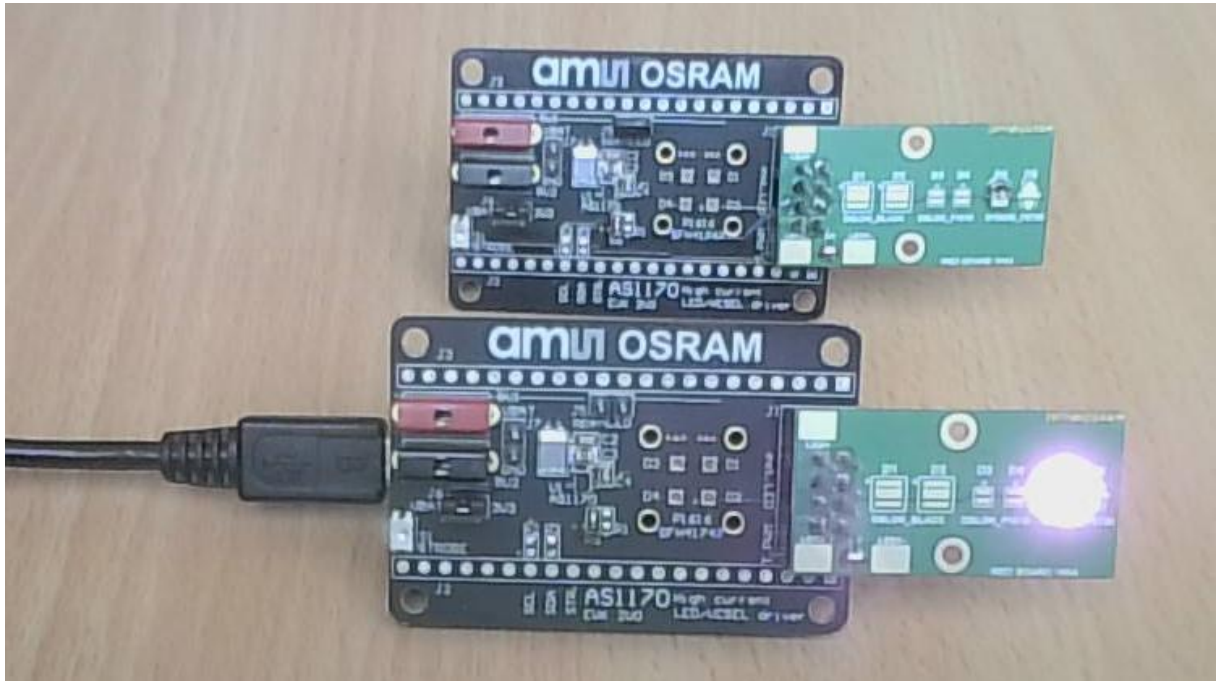
*Note: Don’t exceed 400mA total current via USB connection

Side by Side comparison aO LED with competitor LED

AS1170 indicator mode used like shown on slide 8



aO LED with 400mA and 25% Duty Cycle



Competitor LED with 400mA and 25% Duty Cycle

*Note: Pictures taken with integrated laptop camera

BACKUP Slides
(configuration details for external MCU)

I2C ADDR Selection

I2C Device Address (A-select pin)

EVK Default setting: Pin I2C_ADDR = 0 → I2C address = 0x30 (7-bit) → R6 = 0R & R1 = NC

Alternative setting: Pin I2C_ADDR = 1 → I2C address = 0x32 (7-bit) → R1 = 0R & R6 = NC

Settings:

Set I2C_ADDR pin to 0 = 0x30 I2C address //EVK default setting with R6 = 0R and R1 = NC

Read from ADDR 0h → chip id = **b4** //it is recommended to read CHIP ID after power on to confirm correct communication with AS1170

Alternative I2C slave address:

Set I2C_ADDR to 1 = 0x32 I2C address

Read from ADDR 0h → chip id = **b4h**

Strobe function

Strobe input definition via register 07h

AS1170 can be activated with an input via pin STROBE or via I2C command by setting OUT_ON = “1”

Reg 07h, bit 6: “Strobe type”

- **Default setting → level sensitive:** AS1170 is reacting on rising and falling edge of the strobe signal as long the strobe pulse is shorter than the flash timeout timer
- Edge sensitive: AS1170 is reacting on 1st rising edge of the Strobe signal, flash time is controlled with internal flash timeout timer

Reg 07h, bit 7: “Strobe on”

- **Default setting → input enabled:** AS1170 flash is triggered via STROBE input
- Input disabled**:** AS1170 flash is triggered with control register (OUT_on) and it is controlled with internal flash timeout timer

Table 13. Strobe Signalling Register

Addr: 7		Strobe Signalling Register			
		This register defines the flash current reducing and mode for STROBE			
Bit	Bit Name	Default	Access		Description
6	strobe_type	1	R/W		Defines if the STROBE input is edge or level sensitive; see also bit strobe_on (page 27)
				0	STROBE input is edge sensitive
				1	STROBE input is level sensitive
7	strobe_on	1	R/W		Enables the STROBE input
				0	STROBE input disabled
				1	STROBE input enabled in flash mode

Flash example with default setting **Level sensitive** and **Strobe ON**

yellow: STROBE
blue: LED_OUT
red: ILED current
green: Vbat/Vin

Flash mode settings and examples

Current setting & Flash timeout timer calculation

Default Register Address = 0x10

Addr: 6		Control register		
Bit	**Bit name**	**Default**	**Access**	**Bit description**
1:0	*mode_setting*	00	R/W	Define the AS1170 operating mode. **Value** / **Description** 00: Shutdown. 01: **Indicator mode** (or low current mode using PWM) LED current is defined by the 6LSB bits of led_current1 and led_current2 pwm modulated with 31.25 kHz defined by register inct_pwm (1/16...4/16). 10: **Assist light mode** Led current is defined by the 7LSB(2) bits of led_current1 and led_current2. 11: **Flash mode**(3) Led current is defined by led_current1 and led_current2 (out_on and mode_setting are automatically cleared after a flash pulse).
2	*reserved*	X	R	Reserved - don't use, always write 0.
3	*out_on*	0	R/W	Enables the output current sinks (pin LED_OUT1/2). **Value** / **Description** 0: Outputs disabled. 1: Outputs enabled (out_on and mode_setting are automatically cleared after a flash pulse).
4	*auto_strobe*	1	R/W	Enables the auto trigger of flash mode with strobe pin. **Value** / **Description** 0: Single flash / Strobe mode. 1: In strobe mode, there is no need after flash timeout, the mode_setting and out_on bits are not cleared. A new pulse on STROBE pin will retrigger the flash again. There is no need to rewrite the CONTROL register with I²C command.

(1) This register identifies the operating mode and includes an all on/off bit.
(2) The MSB bit of this register not used to protect the LED; therefore, the maximum assist light current = half the maximum flash current.
(3) Default trigger is done via external pin "STROBE".

Single Flash:

Mode setting: mode_setting “11” / auto_strobe = 0

After a flash is triggered register bits “mode_setting” and “out_on” are reset to “0”. Hence to retrigger a new flash using Strobe pin, an I2C command is required to set the mode_setting[1:0] = 3 and out_on=1.

Multiple Flashes: (recommended)

Mode setting: mode_setting “11” / auto_strobe = 1

With auto_strobe = 1 the bits “mode_setting” and “out_on” are not reset at the end of flash mode. Hence multiple flash in Strobe mode can be triggered directly with the Strobe pin, without a need to write to mode_settings and out_on registers. Flash time out timer performs as time limiter

Single Flash with auto_strobe = 1

Same behavior as on the single flash. But no reconfiguration needed after flash was triggered.

Single flash with external strobe

Programming example single flash


```
my_system.SMU.source_voltage("VIN", 3.7, 3 ,current_range = 3)

my_system.SMU.source_voltage("TORCH", 0, 0.005 ,current_range = 0.01) # Torch pin set to 0

#7bit I2C adress = 0x30

i2c_write(my_system, Reg.Strobe_Signalling, 0xC0 ) # strobe_type = 1 / level sensitive , strobe_on = 1 / external STROBE input enabled in flash mode

i2c_write(my_system, Reg.Flash_Timer, 0x00) # flash timeout timer set to 2ms

i2c_write(my_system, Reg.CURRENT_SET_LED1, 0x7F ) # LED current of LED1 is set to ~450mA

i2c_write(my_system, Reg.CURRENT_SET_LED2, 0x7F) # LED current of LED2 is set to ~450mA


i2c_write(my_system, Reg.Control, 0x01b ) # mode setting = 11 / flash mode , out on = 1 , autostrobe = 1 by default
# autostrobe: enables multiple flash pulses

###################################
#waiting for external strobe pulses
```

yellow: STROBE
blue: LED_OUT
red: ILED current
green: Vbat/Vin

Multiple flash pulses with external strobe

Programming example multiple flash pulses


```
my_system.SMU.source_voltage("VIN", 3.7, 3 ,current_range = 3)

my_system.SMU.source_voltage("TORCH", 0, 0.005 ,current_range = 0.01) # Torch pin set to 0

#7bit I2C adress = 0x30

i2c_write(my_system, Reg.Strobe_Signalling, 0xC0 ) # strobe_type = 1 / level sensitive , strobe_on = 1 / external STROBE input enabled in flash mode

i2c_write(my_system, Reg.Flash_Timer, 0x80) # flash timeout timer set to 129ms

i2c_write(my_system, Reg.CURRENT_SET_LED1, 0x7F ) # LED current of LED1 is set to ~450mA

i2c_write(my_system, Reg.CURRENT_SET_LED2, 0x7F) # LED current of LED2 is set to ~450mA


i2c_write(my_system, Reg.Control, 0x01b ) # mode setting = 11 / flash mode , out on = 1 , autostrobe = 1 by default
# autostrobe: enables multiple flash pulses

###################################
#waiting for external strobe pulses
```

yellow: STROBE
blue: LED_OUT
red: ILED current
green: Vbat/Vin

AS1170 example configuration codes

AS1170 configuration code example

Multiple Flash pulses , external Strobe / level sensitive, auto_strobe = 1 (no re-configuration)

AS1170 power on reset (apply VIN)

#7bit I2C adress = 0x30

i2c_write(ADDR 07h, Reg.Strobe_Signalling, 0xC0)	// strobe_type = 1 / level sensitive , strobe_on = 1 / external STROBE input enabled in flash mode
i2c_write(ADDR 05h, Reg.Flash_Timer, 0x80)	// flash timeout timer set to 129ms
i2c_write(ADDR 01h, Reg.CURRENT_SET_LED1, 0x7F)	// LED current of LED1 is set to ~450mA
i2c_write(ADDR 02h, Reg.CURRENT_SET_LED2, 0x7F)	// LED current of LED2 is set to ~450mA
i2c_write(ADDR 06h, Reg.Control, 0x1b)	// mode setting = 11 / flash mode , out on = 1 / auto_ strobe = 1

####################################

#waiting for external strobe pulses	//multiple consecutive strobe pulses possible, no re-configuration needed
	// flash time defined with high time of STROBE signal or 129ms via flash timeout timer

AS1170 configuration code example

Single Flash , external Strobe / level sensitive, auto_strobe = 0 (re-configuration after flash triggered)

AS1170 power on reset (apply VIN)

#7bit I2C adress = 0x30

i2c_write(ADDR 07h, Reg.Strobe_Signalling, 0xC0)	//strobe_type = 1 / level sensitive , strobe_on = 1 / external STROBE input enabled in flash mode
i2c_write(ADDR 05h, Reg.Flash_Timer, 0x23)	//flash timeout timer set to 36ms (default configuration)
i2c_write(ADDR 01h, Reg.CURRENT_SET_LED1, 0x7F)	//LED current of LED1 is set to ~450mA
i2c_write(ADDR 02h, Reg.CURRENT_SET_LED2, 0x7F)	//LED current of LED2 is set to ~450mA
i2c_write(ADDR 06h, Reg.Control, 0x0b)	//mode setting = 11 / flash mode , out on = 1, auto_strobe = 0

####################################

#waiting for external strobe pulse

####################################

i2c_write(ADDR 06h, Reg.Control, 0x0b)	//mode setting = 11 / flash mode , out on = 1, auto_strobe = 0

#waiting for external strobe pulse

Legal information

Copyright & disclaimer

Demo Kits, Evaluation Kits and Reference Designs are provided to recipient on an “as is” basis for demonstration and evaluation purposes only and are not considered to be finished end-products intended and fit for general consumer use, commercial applications and applications with special requirements such as but not limited to medical equipment or automotive applications. Demo Kits, Evaluation Kits and Reference Designs have not been tested for compliance with electromagnetic compatibility (EMC) standards and directives, unless otherwise specified. Demo Kits, Evaluation Kits and Reference Designs are not prepared or tested for eye safety. Unless specified otherwise, the user is solely responsible for all required safety measures and for complying with the applicable statutory requirements and standards. Demo Kits, Evaluation Kits and Reference Designs shall be used by qualified personnel only.

ams-OSRAM AG reserves the right to change functionality and price of Demo Kits, Evaluation Kits and Reference Designs at any time and without notice.

Any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed. Any claims and demands and any direct, indirect, incidental, special, exemplary or consequential damages arising from the inadequacy of the provided Demo Kits, Evaluation Kits and Reference Designs or incurred losses of any kind (e.g. loss of use, data or profits or business interruption however caused) as a consequence of their use are excluded.

ams-OSRAM AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of ams-OSRAM AG rendering of technical or other services.