Application Note

AS7341

SMUX configuration

Content Guide

1 General Description

AS7341 integrates a so-called super multiplexer (SMUX). With the SMUX, it is possible to map all available photo diodes to one of the six available light to frequency converter. (CH0 ADC to CH5 ADC) Every pixel has a multiplexer to map it to one of the engines – this multiplexer can be configured with 3 bits. (0 = pixel disabled / connected to GND; 1 to 6 = ADC 0 to ADC 5)

The figures below show the SMUX pixel ID mapping to every individual diode. In addition to the 4x4 pixel array, flicker detection, NIR and CLEAR diodes three IDs are available for external photodiodes connected to pin GPIO and INT as well as an on chip DARK photo diode (PD covered with black filter). Not mentioned and grey pixel IDs are not used and shall be programed with “0”.

Reading and writing pixel configuration uses the first 20 bytes of the RAM starting at address 00h. For easier usage, the pixel configuration is stored in nibbles within the RAM (4bits per pixel configuration, MSB not used). It is recommended to write the 20 bytes at once and configure all pixels together within one page write command.

Figure 1: Sensor Array

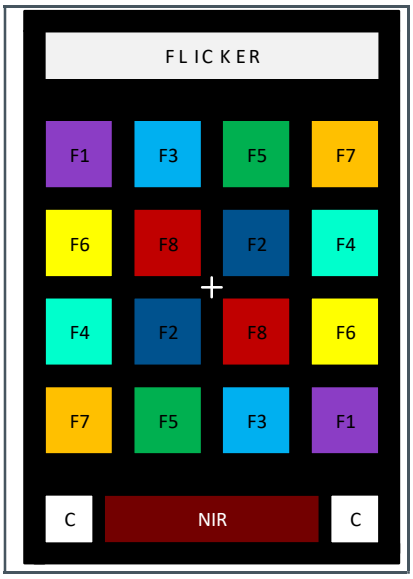


Figure 2: SMUX Pixel ID mapping to Diodes

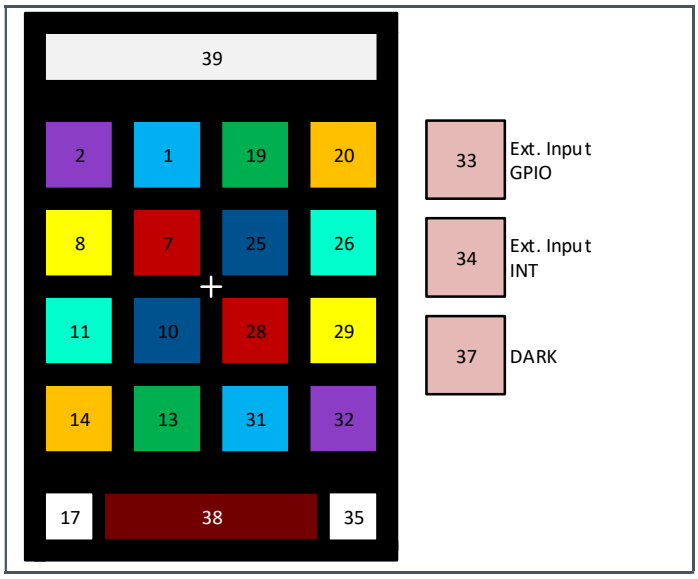
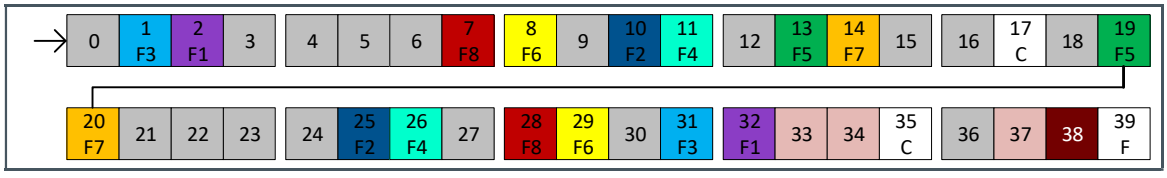


Figure 3: chain map pixel ID vs Filter



2 SMUX multiplexer mapping

The following table shows the mapping of the SMUX pixel IDs to address configuration bit positions in I2C and RAM address space.

I2C ADDR	RAM ADDR	<D7>	<D6>	<D5>	<D4>	<D3>	<D2>	<D1>	<D0>
0x00	0			Pixel ID 1 [6:4]				unused	
0x01				unused				Pixel ID 2 [2:0]	
0x02	1			unused				unused	
0x03				Pixel ID 7 [6:4]				unused	
0x04	2			unused				Pixel ID 8 [2:0]	
0x05				Pixel ID 11 [6:4]				Pixel ID 10 [2:0]	
0x06	3			Pixel ID 13 [6:4]				unused	
0x07				unused				Pixel ID 14 [2:0]	
0x08	4			Pixel ID 17 [6:4]				unused	
0x09				Pixel ID 19 [6:4]				unused	
0x0A	5			unused				Pixel ID 20 [2:0]	
0x0B				unused				unused	
0x0C	6			Pixel ID 25 [6:4]				unused	
0x0D				unused				Pixel ID 26 [2:0]	
0x0E	7			Pixel ID 29 [6:4]				Pixel ID 28 [2:0]	
0x0F				Pixel ID 31 [6:4]				unused	
0x10	8			Pixel ID 33 [6:4]				Pixel ID 32 [2:0]	
0x11				Pixel ID 35 [6:4]				Pixel ID 34 [2:0]	
0x12	9			Pixel ID 37 [6:4]				unused	
0x13				Pixel ID 39 [6:4]				Pixel ID 38 [2:0]	

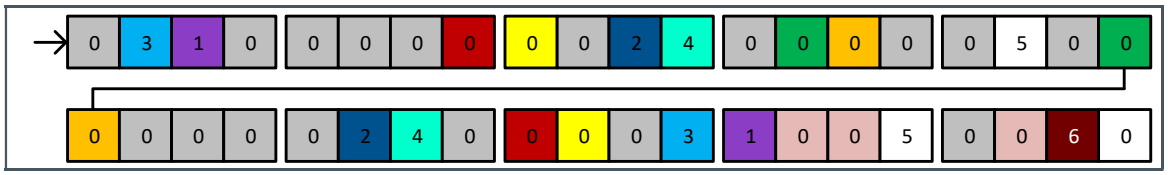
Addr: 0x00 – 0x13		SMUX multiplexer mapping		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	not used	0	RW	reserved
6:4	Pixel IDx	0	RW	0: connected to Ground / disabled 1: Pixel connected to ADC0 2: Pixel connected to ADC1 3: Pixel connected to ADC2 4: Pixel connected to ADC3 5: Pixel connected to ADC4 6: Pixel connected to ADC5 (Flicker) 7: reserved
3	not used	0	RW	reserved
2:0	Pixel IDy	0	RW	0: connected to Ground / disabled 1: Pixel connected to ADC0 2: Pixel connected to ADC1 3: Pixel connected to ADC2 4: Pixel connected to ADC3 5: Pixel connected to ADC4 6: Pixel connected to ADC5 (Flicker) 7: reserved

3 Configuration example read out F1-F4, CLEAR, NIR

The following example shows how to map individual PDs to dedicated ADCs using the SMUX. Each box in the chain map example below represents one nibble (4 bit per pixel ID). The number within the box is the value which needs to be programed in order to map the pixel to the desired ADC.

F1 mapped to ADC0, F2 mapped to ADC1, F3 mapped to ADC2, F4 mapped to ADC3, CLEAR mapped to ADC4 and NIR mapped to ADC5.

Figure 4:
chain map example read out F1 to F4, CLEAR and NIR



Step	I2C command	Description
1	I2C_write(0x80, 0x01)	Enable power (set PON = “1”) and disable SP_EN (SP_EN=”0”) Register: ENABLE / 0x80
2	I2C_write(0xB2, 0x10)	Enable special interrupt (SINT_SMUX). As soon as SMUX command has finished interrupt is activated. Register: CFG9 / 0xB2
3	I2C_write(0xF9, 0x01)	Enable special interrupt SIEN Register: INTENAB / 0xF9
4	I2C_write(0xAF, 0x10)	Write SMUX configuration from RAM to set SMUX chain Register: CFG6 / 0xAF
5	I2C_write(0x00, 0x30)	F3 left set to ADC2
6	I2C_write(0x01, 0x01)	F1 left set to ADC0
7	I2C_write(0x02, 0x00)	
8	I2C_write(0x03, 0x00)	F8 left disabled
9	I2C_write(0x04, 0x00)	F6 left disabled
10	I2C_write(0x05, 0x42)	F4 left connected to ADC3 / F2 left connected to ADC1
11	I2C_write(0x06, 0x00)	F5 left disabled
12	I2C_write(0x07, 0x00)	F7 left disabled
13	I2C_write(0x08, 0x50)	CLEAR connected to ADC4
14	I2C_write(0x09, 0x00)	F5 right disabled
15	I2C_write(0x0A, 0x00)	F7 right disabled
16	I2C_write(0x0B, 0x00)	
17	I2C_write(0x0C, 0x20)	F2 right connected to ADC1
18	I2C_write(0x0D, 0x04)	F4 right connected to ADC3
19	I2C_write(0x0E, 0x00)	F6/F8 right disabled
20	I2C_write(0x0F, 0x30)	F3 right connected to ADC2
21	I2C_write(0x10, 0x01)	F1 right connected to ADC0
22	I2C_write(0x11, 0x50)	CLEAR right connected to ADC4
23	I2C_write(0x12, 0x00)	
24	I2C_write(0x13, 0x06)	NIR connected to ADC5
25	I2C_write(0x80, 0x11)	Start SMUX command while keeping power on (SMUXEN = “1” and PON = “1”)
26		wait for interrupt
27	I2C_write(0x80, 0x00)	power down (PON = “0”)

4 Configuration example read out F5-F8, CLEAR, NIR

The following example shows how to map individual PDs to dedicated ADCs using the SMUX.

F5 mapped to ADC0, F6 mapped to ADC1, F7 mapped to ADC2, F8 mapped to ADC3, CLEAR mapped to ADC4 and NIR mapped to ADC5

Figure 5:
chain map example read out F5 to F8, CLEAR and NIR

Step	I2C command	Description
1	I2C_write(0x80, 0x01)	Enable power (set PON = “1”) Register: ENABLE / 0x80
2	I2C_write(0xB2, 0x10)	Enable special interrupt (SINT_SMUX). As soon as SMUX command has finished interrupt is activated. Register: CFG9 / 0xB2
3	I2C_write(0xF9, 0x01)	Enable special interrupt SIEN Register: INTENAB / 0xF9
4	I2C_write(0xAF, 0x10)	Write SMUX configuration from RAM to set SMUX chain Register: CFG6 / 0xAF
5	I2C_write(0x00, 0x00)	F3 left disabled
6	I2C_write(0x01, 0x00)	F1 left disabled
7	I2C_write(0x02, 0x00)	
8	I2C_write(0x03, 0x40)	F8 left connected to ADC3
9	I2C_write(0x04, 0x02)	F6 left connected to ADC1
10	I2C_write(0x05, 0x00)	F4/F2 disabled
11	I2C_write(0x06, 0x10)	F5 left connected to ADC0
12	I2C_write(0x07, 0x03)	F7 left connected to ADC2
13	I2C_write(0x08, 0x50)	CLEAR connected to ADC4
14	I2C_write(0x09, 0x10)	F5 right connected to ADC0
15	I2C_write(0x0A, 0x03)	F7 right connected to ADC2
16	I2C_write(0x0B, 0x00)	
17	I2C_write(0x0C, 0x00)	F2 right disabled
18	I2C_write(0x0D, 0x00)	F4 right disabled
19	I2C_write(0x0E, 0x24)	F8 right connected to ADC3 / F6 right connected to ADC1
20	I2C_write(0x0F, 0x00)	F3 right disabled
21	I2C_write(0x10, 0x00)	F1 right disabled
22	I2C_write(0x11, 0x50)	CLEAR right connected to ADC4
23	I2C_write(0x12, 0x00)	
24	I2C_write(0x13, 0x06)	NIR connected to ADC5
25	I2C_write(0x80, 0x11)	Start SMUX command while keeping power on (SMUXEN = “1” and PON = “1”)
26		wait for interrupt
27	I2C_write(0x80, 0x00)	power down (PON = “0”)

Buy our products or get free samples online at:
www.ams.com/ICdirect

Technical Support is available at:
www.ams.com/Technical-Support

Provide feedback about this document at:
www.ams.com/Document-Feedback

For further information and requests, e-mail us at:
ams_sales@ams.com

For sales offices, distributors and representatives, please visit:
www.ams.com/contact

Headquarters
ams AG
Tobelbaderstrasse 30
8141 Unterpremstaetten
Austria, Europe

Tel: +43 (0) 3136 500 0
Website: www.ams.com

5 Copyrights & Disclaimer

Information in this document is believed to be accurate and reliable. However, ams AG does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Applications that are described herein are for illustrative purposes only. ams AG makes no representation or warranty that such applications will be appropriate for the specified use without further testing or modification. ams AG takes no responsibility for the design, operation and testing of the applications and end-products as well as assistance with the applications or end-product designs when using ams AG products. ams AG is not liable for the suitability and fit of ams AG products in applications and end-products planned.

ams 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 or applications described herein. No obligation or liability to recipient or any third party shall arise or flow out of ams AG rendering of technical or other services.

ams AG reserves the right to change information in this document at any time and without notice.