Application Note

AS726x

Design Considerations

Content Guide

1 General Description

This Application Note briefly describes AS726x system level design considerations. It includes both UART interface and I²C interface.

2 Hardware Considerations

2.1 UART Connection

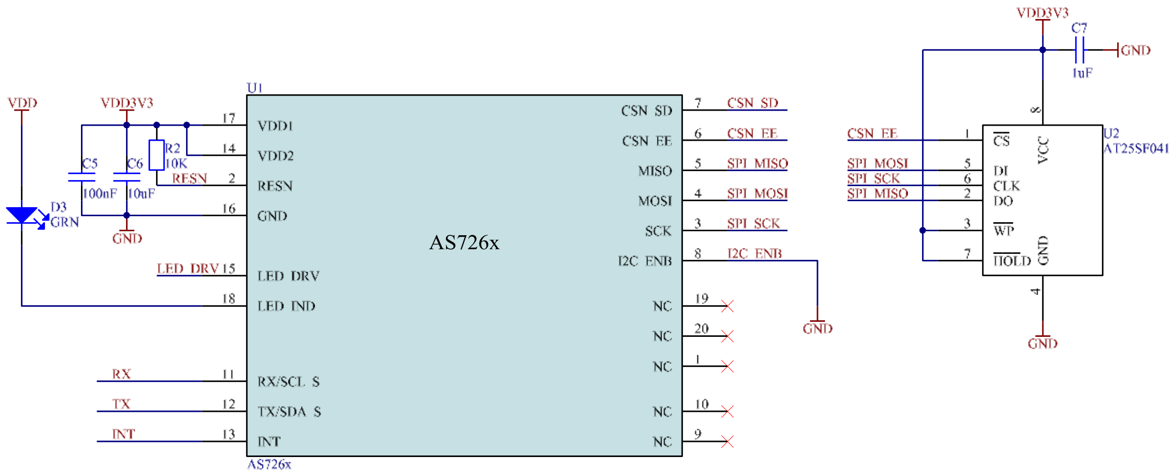
The AS726x sensor provides an UART interface with the pin11, RX and the pin12, TX. These two pins are shared with I²C interface so in order to configure the device with UART interface, the pin8, I²C_ENB, has to be pulled LOW.

AS726x UART interface baud rate is 115200 baud. The data bits is 8-bit, the parity is NONE, and the stop bit is One.

2.1.1 Typical Schematic

The typical schematic in Figure 1 shows AS726x connection with UART interface configuration.

Figure 1. Typical Schematic for UART Interface Configuration



2.1.2 AT Commands

AS726x has various AT commands for device configuration as well as getting the data from the sensor. Any terminal application could be used for AT commands. It is an easier way to get the sensor data. Please refer to the data sheet for supported AT commands.

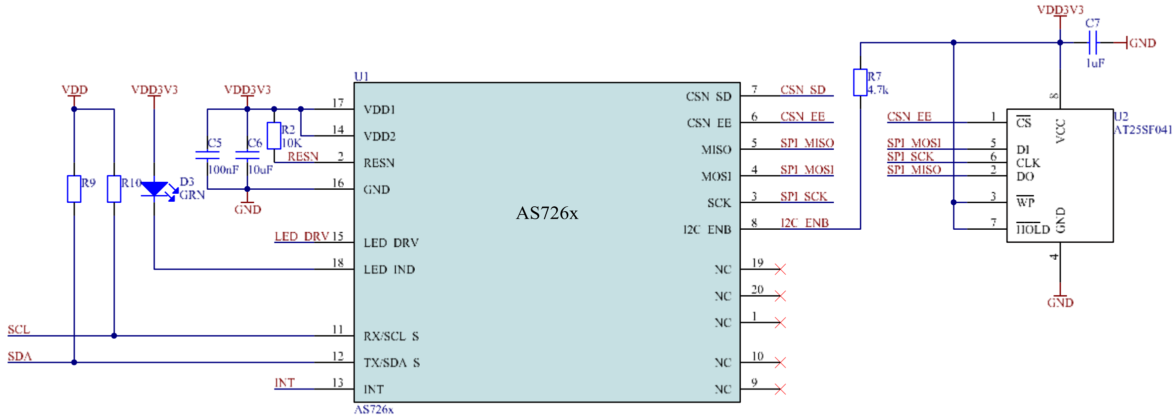
2.2 I²C Connection

The AS726x sensor implements an I²C slave interface with the pin11, SCL_S, assigned to the bus clock and the pin12, SDA_S, for the bus data. These two pins are shared with UART interface so in order to configure the device with I²C bus, the pin8, I²C_ENB, has to be pulled HIGH. Please refer to the resistor R7 in the typical schematic.

2.2.1 Typical Schematic

The typical schematic in Figure 2 shows AS726x connection with I²C interface configuration.

Figure 2. Typical Schematic for I2C Interface Configuration



2.2.2 Pull Up Resistors

According to I²C specification, both SCL and SDA are open drain and need to be connected to a positive supply voltage via a pull-up resistor. The pull-up resistors, R9/R10 in the typical schematic, pull the line high when it is not driven low by the open drain interface. The maximum value of the pull-up resistor is limited by the bus capacitance, C_b, and the rise time, t_r, as below.

$$R_{P(max)} = \frac{t_r}{(0.8473 * C_b)}$$

The bus capacitance is the total capacitance of wire, connections, and pins. I²C Bus specifies the maximum rise time is 300ns.

On the other hand, the minimum value of the pull-up resistor depends on the device logical specifications and allows V_{OL} level to be read as a valide logical low.

$$R_{P(min)} = \frac{V_{DD} - V_{OL(max)}}{I_{OL}}$$

For the AS726x application with 3.3V supply voltage, 0.4V maximum V_{OL}, and the specified minimum sink current of 3mA for standard mode (100KHz) or fast mode (400KHz), the minimum pull-up resistor value is 966.7Ω.

Then the decision of the pull-up resistor value would be based on the rise time, the total bus capacitance, and the power budget. A smaller resistor may get short rise time but has higher power consumption.

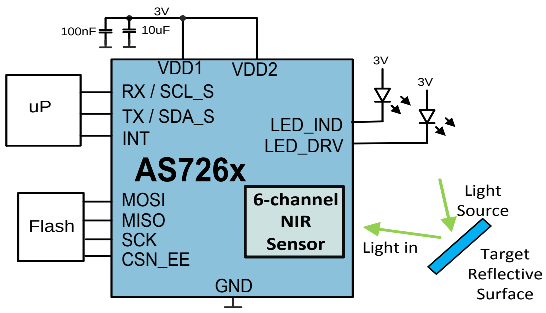
2.3 Power Connection

The AS726x device requires a 3.3V supply on both VDD1 and VDD2 pins associated with the decoupling capacitors, C5/C6 in the schematic. Please note if AS726x device is configured as UART interface, the specification of the supply voltage should be 3.3V±10%. If AS726x is configured as I²C interface, the specification of the supply voltage would be in the range of 2.7V – 3.6V.

2.4 Light Source

2.4.1 Light Source Selection

Figure 3. Typical Application



A typical AS726x application is to shine light rays to a target and AS726x produces outputs based on the reflected light rays as Figure 3. So light source selection is determined by the spectral responsivity of reflected light and characteristics of the target. For example, if the target is expected to absorb 680nm light and the application needs to distinguish the target from others, a broadband white LED might be used as the light source with AS7263 and the 680nm channel should be checked. Various applications may require different light sources.

2.4.2 Light Source Control

AS726x implements two LED drivers. One has the programmable output of 1mA, 2mA, 4mA, or 8mA, which can be used for indication LED. The other one can drive LEDs with the output current 12.5mA, 25mA, 50mA, or 100mA.

AS726x can also work with external LEDs.

2.5 Flash Memory

The AS726x device needs a flash memory to store the firmware and/or persistent data. The flash memory should be 4-Mbits operating at SPI mode 0. AT25SF041 from adesto Technologies and MX25L4006E from Macronix International Co. can be used with AS726x.

3 PCB Layout Considerations

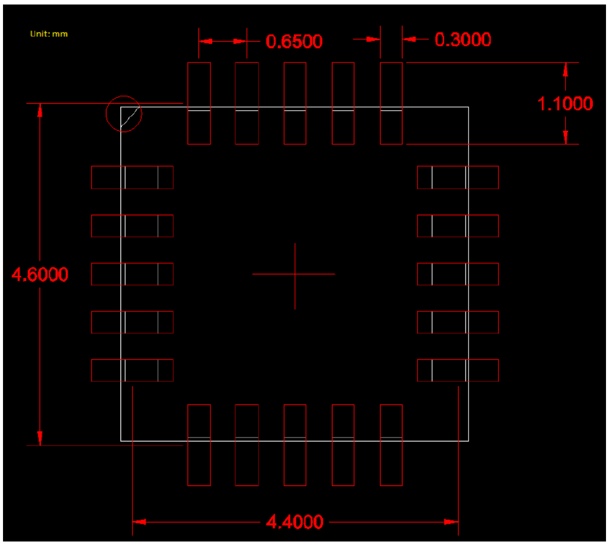
The generic PCB layout rules for digital designs should apply. In general, the wiring must be chosen so that crosstalk and interference to/from the bus lines is minimized. The I²C bus specification also recommends that place VDD and/or GND between SDL and SDA if the traces are longer than 10cm.

The length of I²C bus depends on the load of the bus and the speed you run at. The I²C bus specification defines the maximum capacitance of the bus is 400pF. This bus capacitance limit is specified to limit rise time reductions and allow operating at the rated frequency. In general, with lower frequency and/or lower capacitance of the bus, you can have longer bus length.

For most of I²C bus designs, the capacitance limit should be not the problem at all. If you design involves some unusual conditions, the specification has several strategies to cope with excess bus capacitance. For example, higher drive outputs, bus buffers, switched pull-up circuit etc. Please refer to the specification Section 7.2.

3.1 Footprint

Figure 4. AS726x PCB Footprint Recommendation

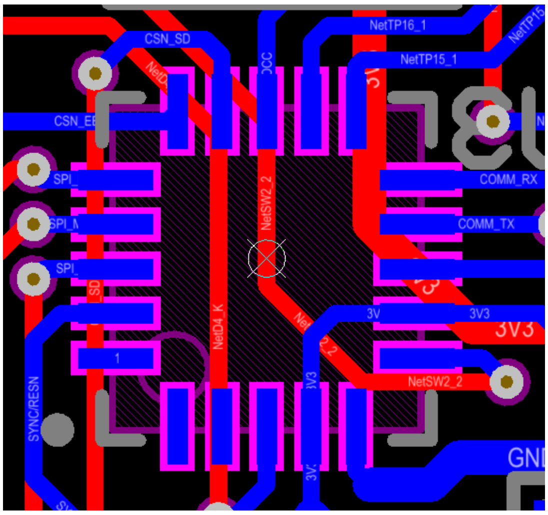


ams provide the schematic symbol and PCB layout footprint in Altium Design format. Please contact **ams** support team to get the library file.

3.2 PCB Layout Considerations

The recommendations to the layout are to place the decoupling capacitors C5/C6 closed to VDD pins of AS726x device and to avoid to put any via underneath the device. Please refer to the Figure 5.

Figure 5. Sample Layout of AS726x Device



3.3 LGA Population Recommendation

Please install the device align with the pads well. Also please make sure all pins would be firmly soldered on the pads.

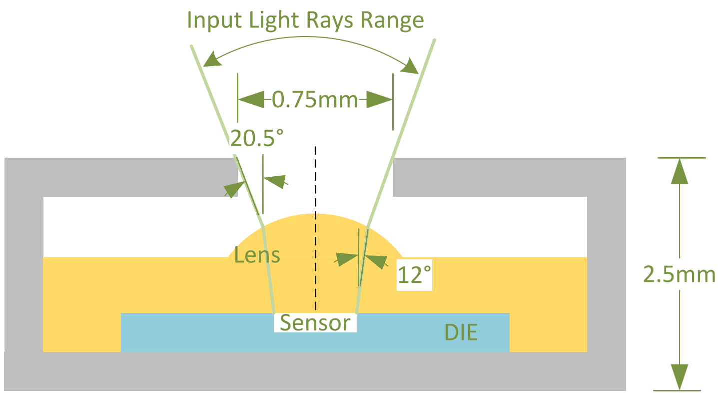
3.4 Optic Considerations

AS726x has an open aperture on the surface. The diameter is 0.75mm and the package field of view is ±20.5°. In order to get accurate sensor data, the angle of the incident light should be carefully designed so that enough light would fell into the range of field of view. On the other hand, the amount of light should be controlled well to avoid the sensor situation.

Since there are six photo diodes in each AS726x device, the incident light rays on each photo diodes should be even. For most reflection applications, the target may diffuse light well enough. If not, extra diffuser may be used between the target and AS726x device.

As an open-aperture device, precautions must be taken to avoid particulate or solvent contamination as a result of any manufacturing processes, including pick and place, reflow, cleaning, integration assembly and/or testing.

Figure 6. Aperture with FOV



4 Software Design Consideration

In most of system designs, AS726x is controlled by a microcontroller. The software of microcontroller design should satisfy both I²C specification and AS726x register structure.

4.1 Features and Register Structure

AS726x supports both standard mode and fast mode. The slave address is 0x49 and the addressing mode is 7+1-bit so when the controller send a read command to AS726x, the salve address and R/W bit should be 0x93 and when sending the write command, it should be 0x92. Both read and write are single byte process.

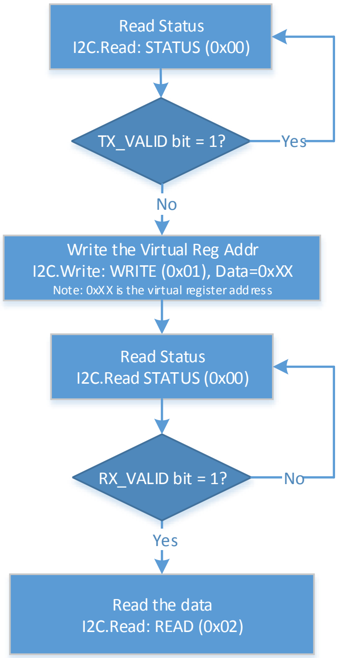
AS726x does not support the slave clock stretching mode.

AS726x has only three hardware based registers, STATUS (0x00), WRITE (0x01), and READ (0x02). The rest are implemented as virtual registers in the firmware. All virtual registers are accessed through WRITE and/or READ registers. Please refer to the data sheets for complete set of virtual registers.

4.2 I²C Virtual Register Read

To read an I²C virtual register, please follow the flow chart below.

Figure 8. Flow Chart for Virtual Register Read



To poll the STATUS register, the controller should write the STATUS address then following a read command to get the value of the STATUS register. The Figure 9 shows the format of the command for polling the STATUS register.

Figure 9. Command for Polling the STATUS Register

Start	0x92	STATUS	Ack	Repeat Start	0x93	Data	Nack	Stop

To write the virtual register address, please program WRITE register with the virtual register address as the following format.

Figure 10. Command for Writing the Virtual Register Address for Reading

Start	0x92	WRITE	Ack	Virtual Reg Addr	Ack	Stop

Finally below is the reading command to get the data.

Figure 11. Command for Reading the READ register

Start	0x92	READ	Ack	Repeat Start	0x93	Data	Nack	Stop

Figure 12. Sample Code of Reading a Virtual Register

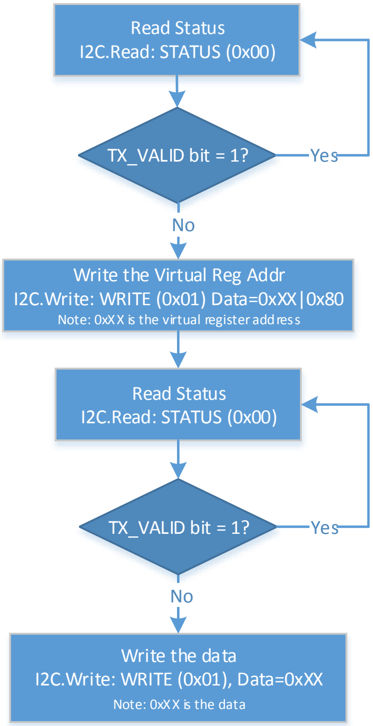
```
#define I2C_AS72XX_SLAVE_STATUS_REG     0x00
#define I2C_AS72XX_SLAVE_WRITE_REG      0x01
#define I2C_AS72XX_SLAVE_READ_REG       0x02
```

```
#define I2C_AS72XX_SLAVE_TX_VALID      0x02
#define I2C_AS72XX_SLAVE_RX_VALID      0x01

uint8_t i2cm_AS72xx_read(uint8_t virtualReg)
{
	volatile uint8_t status, d ;

	while (1)
	{
		// Read slave I2C status to see if we can write the reg address.
		status = i2cm_read(I2C_AS72XX_SLAVE_STATUS_REG) ;

		if ((status & I2C_AS72XX_SLAVE_TX_VALID) == 0)
			// No inbound TX pending at slave.  Okay to write now.
			break ;
	}
	// Send the virtual register address
	i2cm_write(I2C_AS72XX_SLAVE_WRITE_REG, virtualReg) ;

	while (1)
	{
		// Read the slave I2C status to see if our read data is available.
		status = i2cm_read(I2C_AS72XX_SLAVE_STATUS_REG) ;

		if ((status & I2C_AS72XX_SLAVE_RX_VALID) != 0)
			// Read data is ready for us.
			break ;
	}
	// Read the data to complete the operation.
	d = i2cm_read(I2C_AS72XX_SLAVE_READ_REG) ;
	return d ;
}
```

4.3 I²C Virtual Register Write

Writing to a virtual register is similar to the read.

Figure 13. Flow Chart for Virtual Register Write



Please refer to the previous section for polling the STATUS register.

Writing the virtual register address for writing is not same as the one for reading. The MSB of the virtual register address has to be set to 1 for writing.

Figure 14. Command for Writing the Virtual Register Address for Writing

Start	0x92	WRITE	Ack	Virtual Reg Addr | 0x80	Ack	Stop

Simple command for writing the data as below.

Figure 15. Command for Writing the Data

Start	0x92	WRITE	Ack	Data	Ack	Stop

Figure 16. Sample Code of Writing a Virtual Register

```
void i2cm_AS72xx_write(uint8_t virtualReg, uint8_t d)
{
        volatile uint8_t    status;

        while (1)
        {
                // Read slave I2C status to see if we can write the reg address.
                status = i2cm_read(I2C_AS72XX_SLAVE_STATUS_REG);

                if ((status & I2C_AS72XX_SLAVE_TX_VALID) == 0)
```

```
                // No inbound TX pending at slave.  Okay to write now.
                break ;
    }
    // Send the virtual register address
    // (setting bit 7 to indicate a pending write).
    i2cm_write(I2C_AS72XX_SLAVE_WRITE_REG, (virtualReg | 0x80)) ;

    while (1)
    {
        // Read the slave I2C status to see if we can write the data byte.
        status = i2cm_read(I2C_AS72XX_SLAVE_STATUS_REG) ;

        if ((status & I2C_AS72XX_SLAVE_TX_VALID) == 0)
                // No inbound TX pending at slave.  Okay to write data now.
                break ;
    }
    // Send the data to complete the operation.
    i2cm_write(I2C_AS72XX_SLAVE_WRITE_REG, d) ;
}
```

5 Contact Information

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

Tobelbader Strasse 30

8141 Premstaetten

Austria, Europe

Tel: +43 (0) 3136 500 0

Website: www.ams.com

6 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.

7 Revision Information

Changes from previous version to current revision 1-01 (2017-Apr-13)	Page
Updated UART, I^2C Schematic	3, 4
Added Light Source Section	5
Updated Optic Consideration Section	7

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