anitoa

Anitoa ULS24 Datasheet

Version 0.60

Last updated: October 13, 2016

Draft

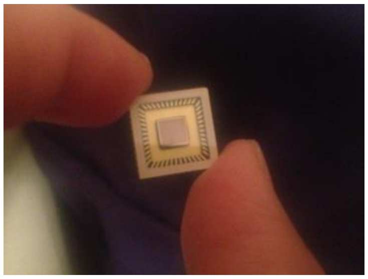
Contents

©2014-2016, Anitoa Systems, LLC

Draft

Draft

Chapter 1 General description



The Anitoa ULS24 is an ultra-low-light image sensor. Its low cost, small form factor and high level of integration make it optimally suited for use in a portable device in medical, scientific and industrial applications. An example of such application is a field portable nucleic-acid-test (NAT) system that performs molecular sensing based-on fluorescence or chemiluminescence signaling principles.

ULS24 is built on 0.18um CMOS process at a world-leader specialty semiconductor foundry. ULS24 has 24 x 24 pixels.

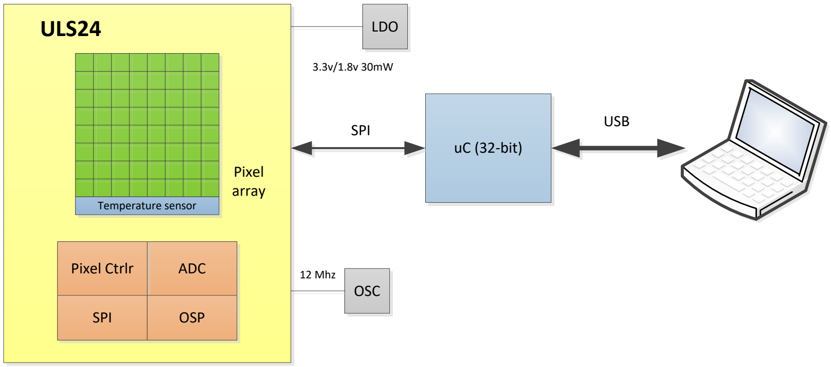


Figure 1.1 ULS24 Chip Application Block Diagram

©2014-2016, Anitoa Systems, LLC

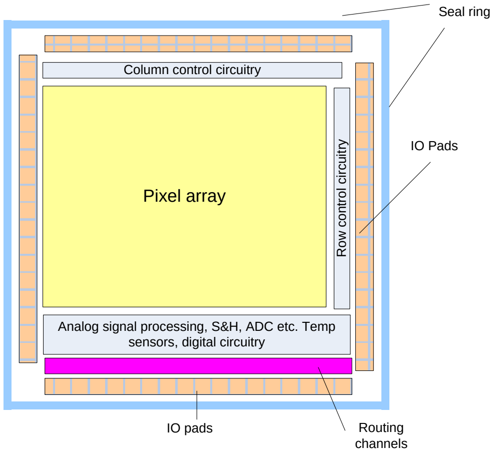


Figure 1.2 ULS24 chip layout

1.1 Features

- Ultra low-light sensitivity. Detection threshold ~3.0 x 10^{-6} lux[1]
- Low dark current, high SnR (>13dB at detection threshold).
- 12-bit ADC
- Wide dynamic range (> 85dB). Full well at 800Ke in high gain mode and 4.9Me in low gain mode. Good linearity (<0.5% low gain mode; <0.6% high gain mode)
- Digital interface through Serial Peripheral Interface (SPI).
- Built-in junction temperature sensor.
- 3.3V and 1.8V power supply, 30mW max power.
- Operating temperature range -15 ºC – 85 ºC[2].

[1] Assuming 4 second integration time

[2] The image sensor can meet noise spec at junction temperature up to 55 ºC

©2014-2016, Anitoa Systems, LLC

1.2 Key parameters

Imager size	4.9mm x 4.8mm; Sensing area: 3.6mm x 3.6mm
Active pixels	24x24
Pixel size	150um x 150um (4 can be combined into a 300um x 300um “big pixel”
Integration time	100us - 100s, software controlled
ADC resolution	12-bit
Signal Interface	Serial Peripheral Interface (SPI), 4 wires
Responsivity	135V/lux-sec @ 550nm in high gain mode 540V/lux-sec @ 550nm in high gain 4-bin mode 4200 V/lux-sec with 2850k light in high gain 4-bin mode
Detection threshold	~3.0 x 10^{-6} lux @ 550nm in high gain 4-bin mode
Dynamic range	>85dB
Non-linearity error	Low gain mode: <0.5%; high gain mode: <0.6%
SnR	13dB minimum at detection threshold
Supply	3.3V (analog); 1.8V (digital)
Power consumption	< 30mW
Operating temperature	-15 – 85 ºC (up to 45 ºC to meet optical performance spec).
Temperature sensor spec	±0.3 ºC accuracy. -15 to 85 ºC range.
Packaging	48-pin CLCC

1.2.1 Quantum efficiency as function of wavelength

The curve below shows the quantum efficiency (QE) x fill factor (FF) of ULS24 as a function of the incident light wavelength. Since the filler factor of ULS24 pixels is nearly 100%, the curve can also be interpreted as QE vs. Wavelength relationship.

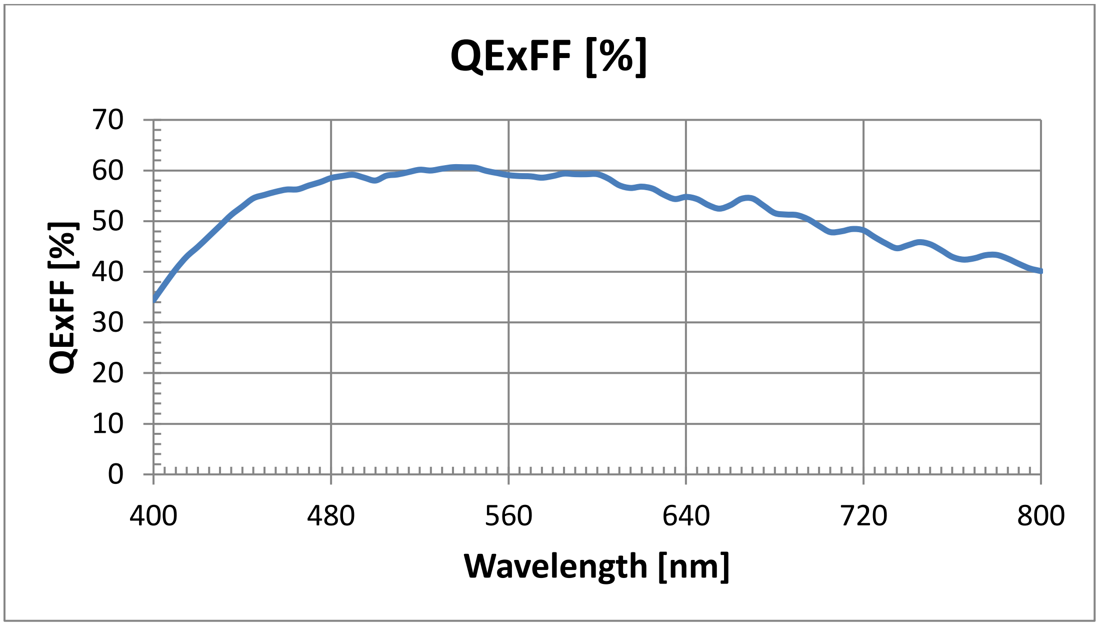


Figure 1.3 QExFF vs. Wavelength

©2014-2016, Anitoa Systems, LLC

1.3 Applications

Biomedical and life science

Molecular sensing/imaging based fluorescence and chemiluminescence signaling principles.

- DNA and RNA quantification, fluorescent-based
- Miniaturized qPCR system or digital qPCR
- Fluorescence or chemiluminescence-based Immunoassay/ELISA
- DNA or Protein microarray
- Pyro-sequencing
- Capillary electrophoresis
- Cell sorting/Imaging flow cytometry
- Fluorescence Images Guided Surgery (FIGS)

Industrial and scientific

- Quantum dot spectrometer
- Hazard and chemical-threat detection instruments

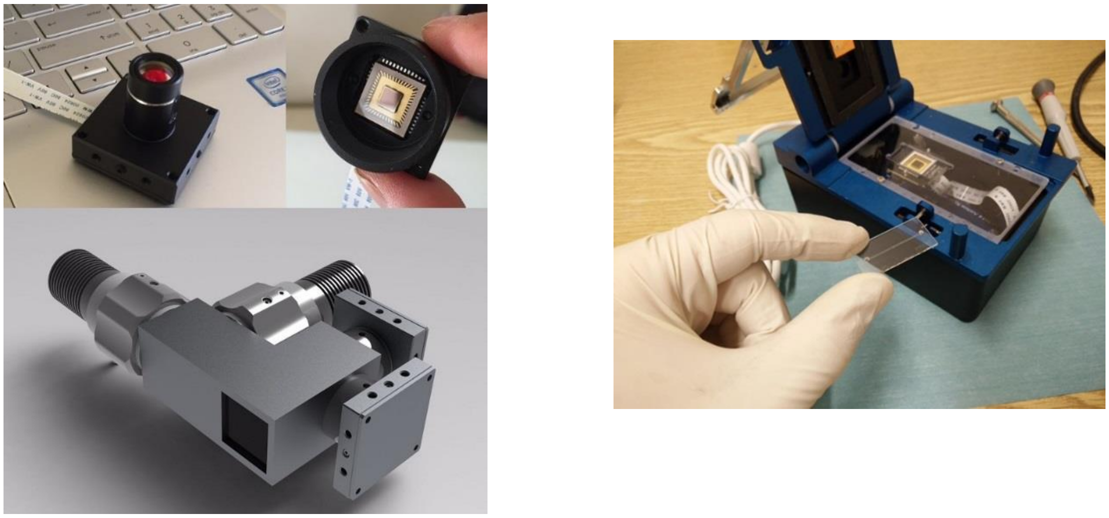


Figure 1.4. ULS24-based fluorescence and chemiluminescence cameras. Left top: 1-channel system fluorescent imager with ULS24, with filter installed; and ULS24 with C-mount lens interface for chemiluminescence imaging. Left bottom: 2 channel imager with integrated 2

channel UV excitation LED for tissue imaging. Right: ULS24 used in a lensless configuration for chemiluminescence read out in a microfluidic application.

1.4 Development kits and modules

The ULS24 Solution Kit is available as an “out of box” solution to allow users to conveniently evaluate ULS24. The ULS 24 Solution Kit includes the ULS 24 Sensor IC, an interface board, and software. This solution kit can readily interface with a PC via USB.

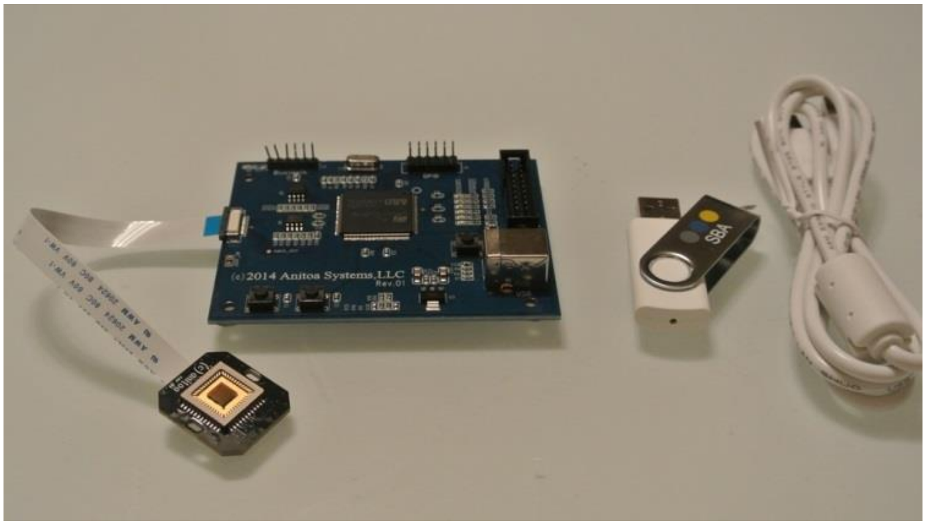


Figure 1.5 ULS24 Solution Kit

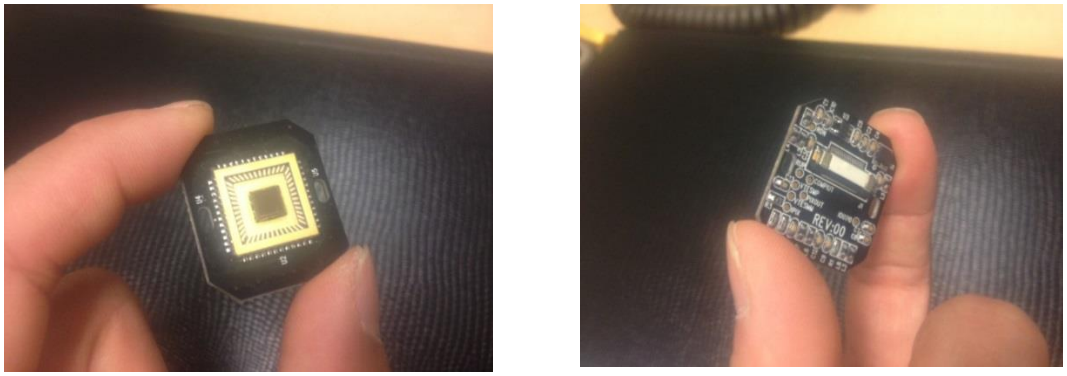


Figure 1.6 ULS24 Sensor Module

Chapter 2 Technical specification

2.1 Pin list

Below is a list of all the pins of ULS24. Some of these pins are for debug only. These debug-only pins are intended for use exclusively for factory testing. Detailed description of the debug only pins is not included in this table.

Name	Pin#	Digital/ Analog /Power	In/Out	Voltage class	Notes
ADC_CLK	10	D	In	3.3V	12Mhz ADC Clock
ADC_RDY	8	D	Out	3.3V	ADC Ready indicator
SSEL	5	D	In	3.3V	SPI Slave select
MOSI	6	D	In	3.3V	Master out slave in
MISO	9	D	Out	3.3V	Master in slave out
SCK	7	D	In	3.3V	SPI clock
AVDD	1,15, 19,45	P	Power	3.3V	3.3V Analog Vdd
AGND	16,20, 44,48	P	Power	3.3V	Analog Ground
AVDDO	17,36, 47	P	Power	3.3V	3.3V Analog VddIO
AVSSO	18,35, 46	P	Power	3.3V	Analog VddIO Ground
VDDA_Pix	38	P	Power	3.3V	3.3V Pixel Vdd
GNDA_Pix	37	P	Power	3.3V	Pixel Ground
VDDIO	11,27, 39	P	Power	3.3V	Digital IO Vdd
VSSIO	12,28, 40	P	Power	3.3V	Digital IO Gnd
VDD	13,29, 41	P	Power	1.8V	1.8V Digital Vdd
VSS	14,32, 42	P	Power	1.8V	Digital Ground
Tempoutp	3	A	Out	3.3V	Temperature sensor output

©2014-2016, Anitoa Systems, LLC

Draft

Tempoutm	4	A	Out	3.3V	Temperature sensor output
Amux out	2	A	Out	3.3V	Test analog signals
Iout10	43	A	Out	3.3v	Current source monitor pins: 10uA
IPIX	33	A	Out	3.3v	Pixel current source monitor pins: 2-14uA
POR (Reset)	26	D	In	3.3V	Power on reset

2.2 Absolute Maximum Ratings

AVDD, AVDDO, VDDA-Pix, VDDIO	-0.3V to 4V
All other 3.3V domain signal pins	-0.3V to 4V
VDD 1-3	-0.3V to 2.4V
Operating temperature	-40 ºC to 125 ºC
Junction temperature	125 ºC

2.3 Electric characteristics

Symbol	Parameter	Min	Typical	Max	Unit
AVDD	Analog supply	3	3.3	3.6	V
AVDDO	Analog IO supply	3	3.3	3.6	V
VDDA-Pix	Pixel Analog supply	3	3.3	3.6	V
VDDIO	Digital IO supply	1.62	3.3	3.6	V
VDD	Digital power supply	1.62	1.8	1.98	V
ADC_Clk	ADC Clock		12		MHz
I_Supply	Total chip supply current (ADC_Clk: 12Mhz, SCK: 1MHz)	8.1	9	9.9	mA
SCK	SPI Clock	DC	1	10	Mhz

2.4 Pixel optical properties

(Tj = 25 ºC; AVDD = 3.3V; VDD = 1.8V; ADCClk = 12MHz; SPI SCK = 1MHz)

Symbol	Parameter	Min	Typical	Max	Unit
Pixel pitch	Pixel pitch		150		um
Spectral response		200		1100	nm
QE X FF (@550nm)	Quantum efficiency x fill factor, @550nm		65		%
Responsivity*	@ 550nm single wavelength light, high gain mode		135		V / lux*sec
Responsivity*	@ 2850k broad spectrum, high gain mode		1050		V/lux*sec
Responsivity*	@ 550nm single wavelength light, low gain mode		17		V / lux*sec
Responsivity*	@ 2850k broad spectrum, low gain mode		132		V/lux*sec
Dark current	At high gain mode		3.4		mV/s
Dark current	At low gain mode		0.4		mV/s
Full well	Full well capacity in Volts		2.0		V
Full Well	Full well capacity in e- (high gain) (low gain)		0.8 4.9		Me-
Readout noise	Total read out noise in dn rms.		2.5		digital unit
Readout noise	Total read out noise equivalent light level @550nm		1.67 x 10^{-7}		lux
PRNU			0.5	1.0	%
ADC resolution	Resolution of the ADC		12		bit

*In 4-bin mode, the responsivity is x4. E.g. responsivity is 4200V/lux-sec in 4-bin mode and with 2850k broad spectrum light.

2.5 Imager electric-optical performance

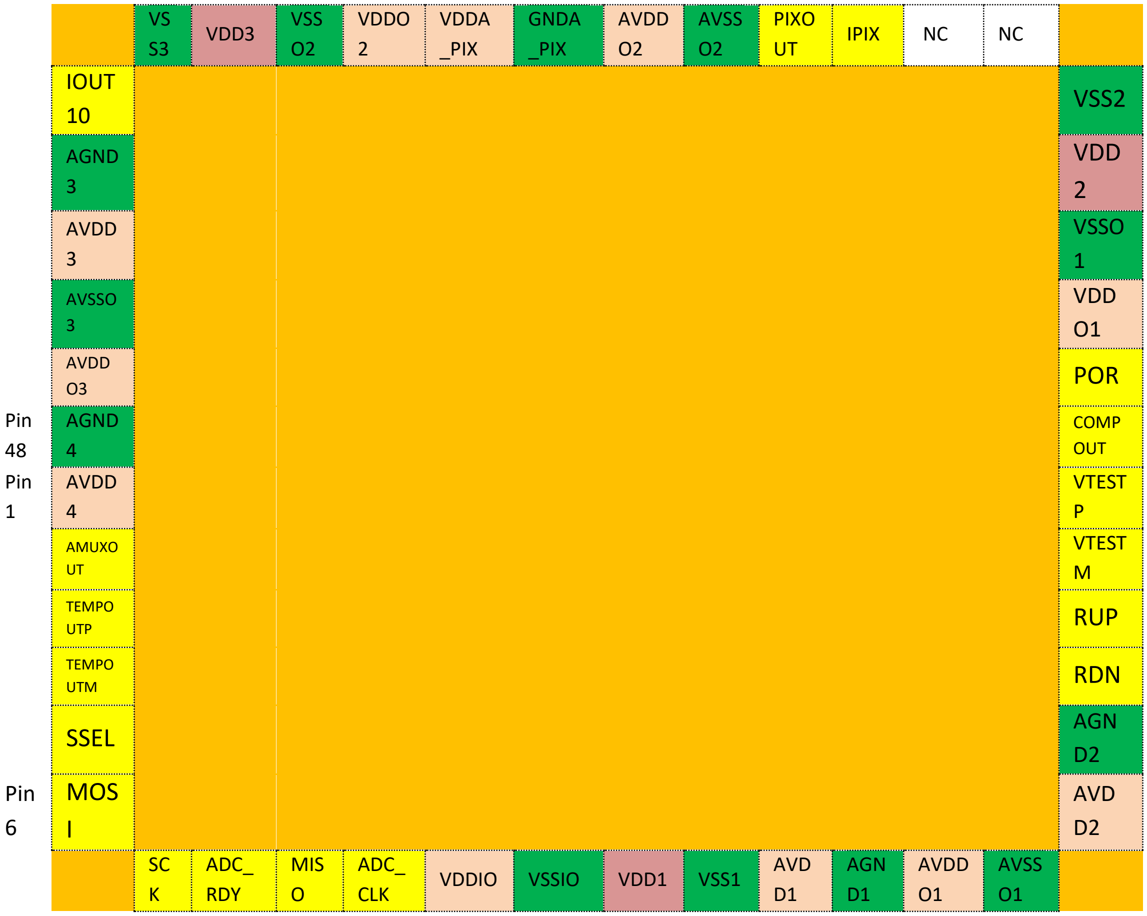
(Tj = 25 ºC; AVDD = 3.3V; VDD = 1.8V; ADCClk = 12MHz; SPI SCK = 1MHz)

Symbol	Parameter	Min	Typical	Max	Unit
Integration time		0.0001		100	s
Frame rate				900	Frame/s
Detection threshold*	Minimum light level to be detected, @550nm		3.0x10^{-6}		lux
SnR*	Signal to noise ratio at detection threshold	13	15		dB
Dynamic range	Intra-scene dynamic range		85		dB
Linearity**	Non-linearity error, low gain mode			0.5	%
Linearity**	Non-linearity error, high gain mode			0.6	%
Readout and data transit time			0.83	1.0	ms/frame

* Based on 4s integration time; high gain and 4-bin mode.

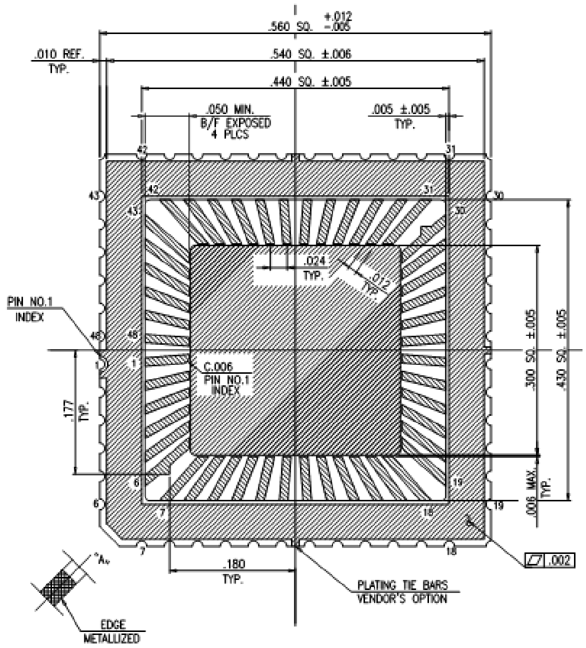
** Based on EMVA1288 testing standard.

2.6 Pin arrangement on package level: (Top View)

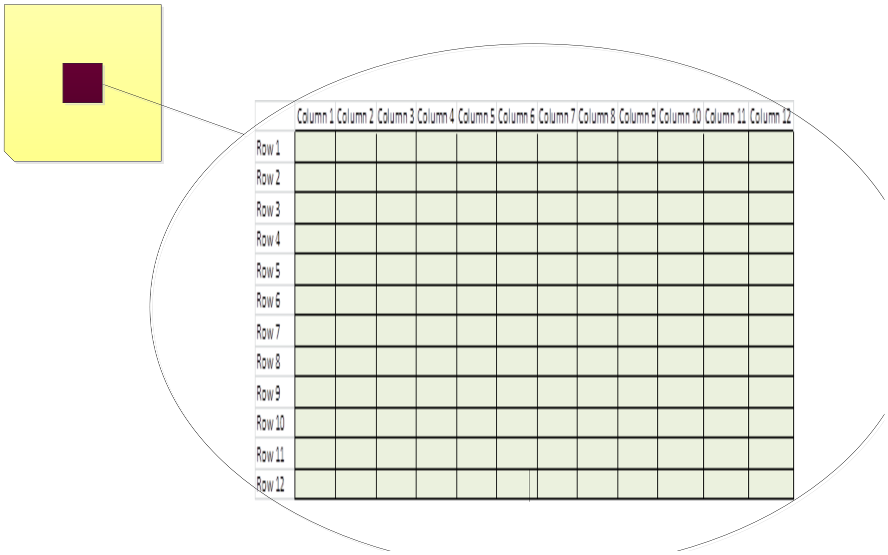


		VSS3	VDD3	VSSO2	VDDO2	VDDA_PIX	GNDA_PIX	AVDDO2	AVSSO2	PIXOUT	IPIX	NC	NC	
	IOUT10													VSS2
	AGND3													VDD2
	AVDD3													VSSO1
	AVSSO3													VDDO1
	AVDDO3													POR
Pin 48	AGND4													COMPOUT
Pin 1	AVDD4													VTESTP
	AMUXOUT													VTESTM
	TEMPOUTP													RUP
	TEMPOUTM													RDN
	SSEL													AGND2
Pin 6	MOSI													AVDD2
		SCK	ADC_RDY	MISO	ADC_CLK	VDDIO	VSSIO	VDD1	VSS1	AVDD1	AGND1	AVDDO1	AVSSO1	

2.7 Package dimension (Top View[3])



The orientation of the chip relative to the package is shown below.



	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12
Row 1												
Row 2												
Row 3												
Row 4												
Row 5												
Row 6												
Row 7												
Row 8												
Row 9												
Row 10												
Row 11												
Row 12												

[3] Top view means imaging sensing area is facing the reader

2.8 ULS24 die dimension:

Parameter	X	Y	Unit
Chip die area	4880	4750	um
Pixel array area (Top left coordinate)	539.715	4396.5	um
Pixel array area (Bottom right coordinate)	4142.495	762.5	um
Chip die thickness	750		um

Chapter 3 Functional Description

ULS24 is comprised of several major functional blocks. These are:

1. Pixel array
2. Pixel driving circuitry
3. ADC and ADC data buffers
4. SPI interface
5. Temperature sensor
6. Debug interface

3.1 Pixel array

The pixel array is a matrix of active pixel sensors (APS). The function of the pixels is to collect light and convert photons to electric signals.

The ULS24 contains a matrix of 24 x 24 pixels. Every 4 adjacent pixels form a “pixel group”. Thus a 24 x 24 pixel array contains 12 x 12 pixel groups - that is 12 rows and 12 columns of pixel groups. These are labeled row1, row2, row3, … etc.; column1, column2, column 3, … etc.

Pixel2	Pixel1
Pixel8	Pixel4

Figure 3.1. A "pixel group" is comprised of 4 regular pixels

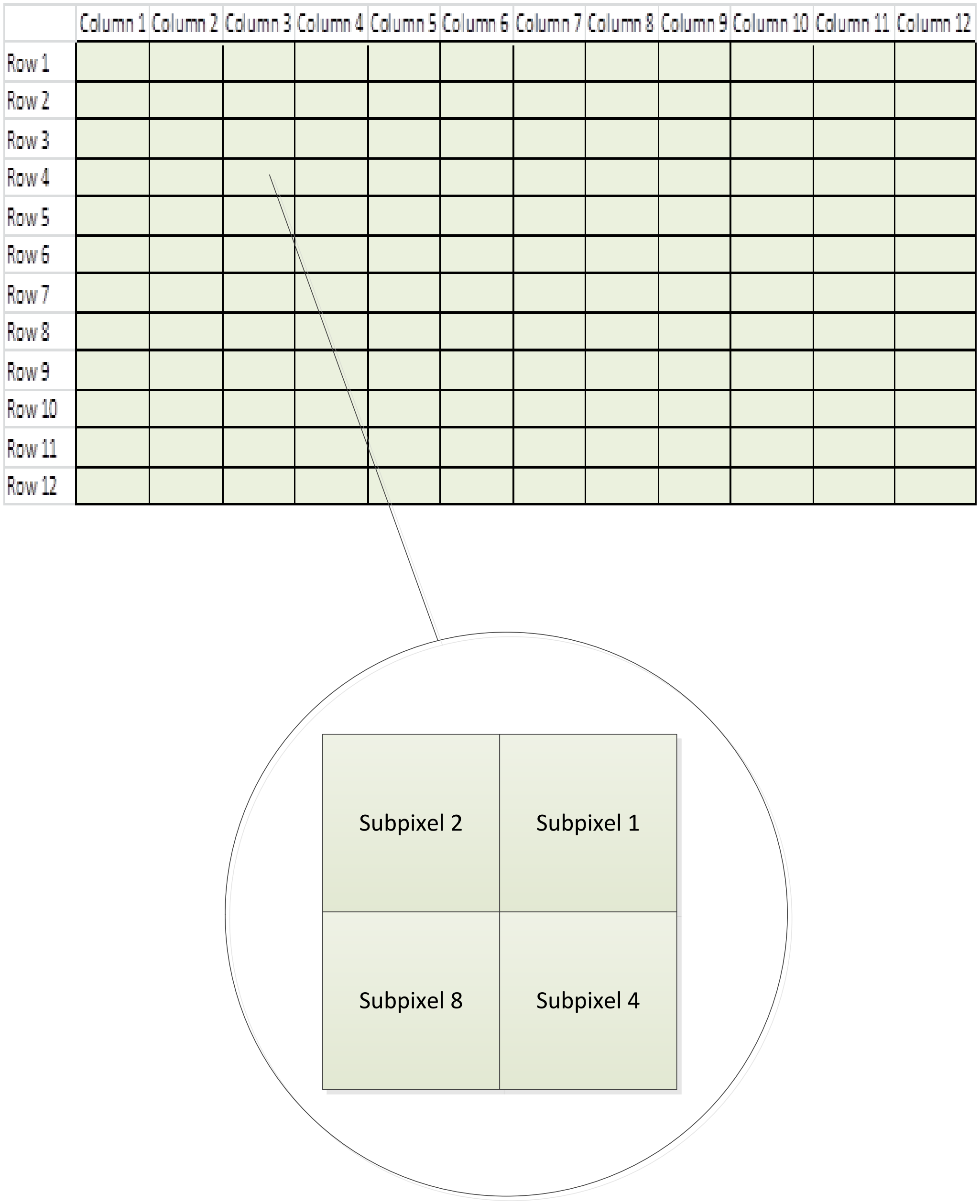


Figure 3.2 The pixels of the imager are arranges as a matrix shown below. Each pixel group is comprised of 4 subpixels.

3.1.1 Relationship between regular pixels and "pixel groups"

As stated above, 4 regular pixels form a pixel group. There is only one read out path from each pixel group. Thus at a given time, only pixel groups can be selected for capture light input. In order for the individual pixels in a pixel group to sense light independently, 4 integration procedures have to be performed sequentially for each pixel in a pixel group to collect light. As a result, 4 integration periods is required for the ULS24 sensor to capture image in 24x24 mode.

When a pixel group is driven to perform light collection, any combination of the 4 pixels in a pixel group can be turned on or off during light sensing integration period. It is therefore possible to combine multiple pixels in a pixel group, and make them act as one pixel in an integration period to increase the light collection efficiency (higher responsivity) and achieve better sensitivity, at the cost of reduced resolution.

3.2 Pixel driving circuitry and special function registers (SFRs)

The pixel driving circuitry is responsible for delivering the necessary driving signals to the pixels to perform reset, integration, and read out.

The output of the pixel is sampled twice in an integration period. The first sample captures the reset voltage level. The second sample captures the output level of the integration capacitor. The two sampled levels are then fed to the differential input of the ADC for conversion to digital signal. This method of sampling is called "correlated double sampling", or CDS. Note CDS is done automatically in hardware.

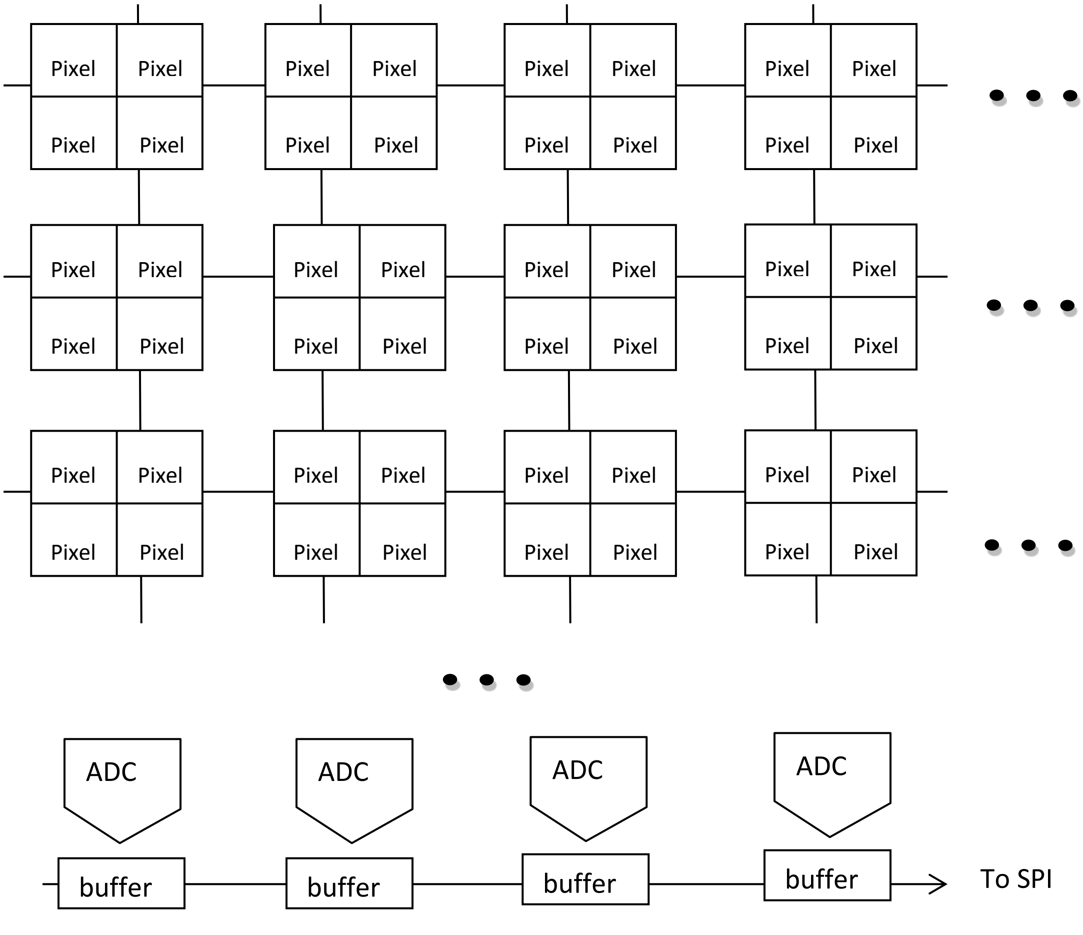


Figure 3.3 Pixel array with associated ADC and data buffer registers

For each column of the pixel groups, there is one ADC and a buffer register to capture the digital output.

At a given time, only one row is selected for receiving pixel driving command inputs. In order to control pixels in row i (i = 1 to 12), the host controller[4] have to first select row i, and then pass the commands to that row. During this time, all the other pixel rows are not receiving any driving signals.

In a typical integration sequence, the controller first sends a reset command. After sending reset command, the controller should wait for some time to let the pixels perform integration. The controller then issues a set of commands to perform read out and data transfer.

The controller can therefore choose the integration time by simply insert a delay between reset command and readout command.

The details of reset command, readout command and data transfer is described in the next few sections.

3.2.1 Special function registers

There are many “Special Function Registers (SFRs)” in ULS24. These are responsible for controlling various aspects of device operation.

The host controller can write to these special function registers, as a means to control the operation of the ULS24 device.

[4] The “ host controller” referenced here is typically an external device, such as a microcontroller or embedded processor.

3.3 The SPI Interface and Control Protocol

The ULS24 has a serial peripheral interface, or SPI. This is a full-duplex serial interface. It is through this interface that a host device, such as a microcontroller, controls the operation of ULS24 and receive the image data.

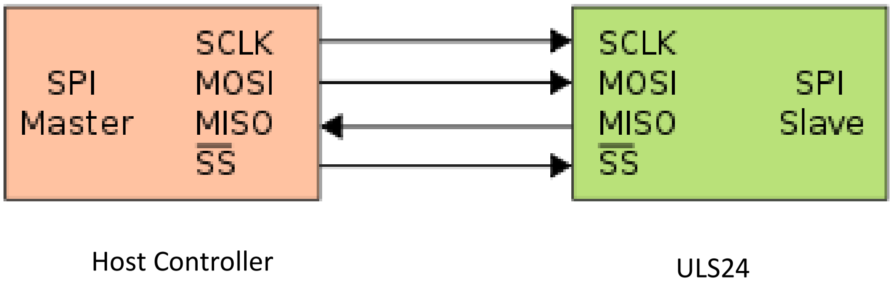


Figure 3.4 Serial Peripheral Interface - SPI

The SPI interface is a 4 wire interface as shown above, between a master and a slave. The ULS24 is the SPI slave. The host is generally a microcontroller.

In the SPI standard, there are several variations of the SPI interface. The ULS24 adopts the convention such that the clock is active high. The data is sampled at rising edge of the clock, and toggles at the falling edge of the clock. This is the most commonly used SPI clocking scheme and is universally supported by microcontrollers on the market.

©2014-2016, Anitoa Systems, LLC
Draft

3.4 SPI interface protocol for ULS24

3.4.1 The host-receive-ULS24-transmit (read) path

The SPI operation is full duplex, as long as the host controller (master) pulls SPI.SS low and issue SPI.SCLK, the host controller is always receiving data from the ULS24's image data buffer.

The ULS24 data buffer is a 24-byte (that is 12 x 16-bit) shift register. There are two 8-bit registers (total 16-bit) for each columns of pixel groups. Each of the pairs of 8-bit registers contains the low and high byte results from ADC operation for the pixel column.

The SPI interface, when enabled and clocked, is constantly shifting out data from the data buffer register to the master. (However, the SPI interface does not care whether the data from the image data buffer is valid or not. (See Chapter 5, the host controller needs to check "ADC Rdy signal to ensure the data is valid). For each of the 12 columns of pixel group, 2 bytes (16-bit) of data are shifted out sequentially. The low order byte is shifted out first. The high order byte second.

The data from the buffer registers of the 12 columns of pixel groups are shifted out sequentially from column1 to column2, column3, ..., column 12, low byte first followed by high byte. There is an internal pointer that points to the data buffer register that is currently shifting data. The value of the pointer increments automatically whenever a byte of data has been shifted out. So this pointer will cycle through registers, such that the data shifted out will be Col1 low byte, Col1 high byte, Col2 Low byte, Col2 High byte, … etc..

After the pointer reaches 23[5], it will continue to increase until 31, then the pointer will wrap around and point to the first register again and continue to cycle through the registers. When slave select signal (SS) is pulled from high to low, the data buffer pointer is reset to 0, regardless of the current value of the pointer. The data shifting will start from the first register again.

[5] The pointer is 0 based. 0: Col1 low byte; 1: Col1 High byte; 2: Col2 Low byte; 3: Col2 High byte etc.

3.4.2 The host-transmit-ULS24 - receive (write) path

The SPI transmission path is used by the host to send commands to the ULS24 as a means to control ULS24 operation.

There are 3 types of command sequences that ULS24 accepts. These are:

1. Pixel driving commands
2. Special function register write commands
3. ADCStart command

The host starts a command sequence by pulling SPI.SS from high to low. After that, the host starts to send command packets consist of a command header byte, followed by one or more command data byte.

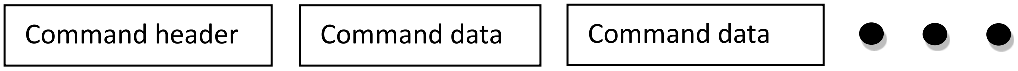


Figure 435 The command sequence

The most significant 3 bit of the command header byte determines the command type.

Command header format

7	6	5	4	3	2	1	0
Command type			Command sub-type				

After a command type is established, while the SPI.SS remain low, the host can send additional command data to drive pixels, or write to special function registers.

3.4.3 Command header definition

Command type	Command sub-type
011 – Pixel driving command	Bits 0 to 3, Selects pixel row 0 to 15
010 – Special Function Register write command	Bits 0 to 2, select registers 0 to 7
100 – Start ADC	N/A

Based on this definition, the command hex codes are:

Start ADC	0x80
Register write	0x40 – 0x47
Pixel command	0x60 - 0x6F

For Pixel driving command, the command header is followed by one or more command data bytes. For SFR write command, the command header is followed by one command data byte. For ADCStart command type, there is no command data

3.4.4 Command data bytes - pixel driving mode

In pixel driving mode, the command data is written to a control register called “Pixel control register”. Pixel control register controls the pixel driving circuitry. Only one row of pixels is selected at a time to receive commands. Selection of pixel row is achieved through the sub-type data contained in pixel driving command header byte.

If the host needs to send commands to another pixel row, it needs to toggle Slave Select (SPI.SS) again and send out another command header byte with a different row selection.

3.4.5 Command data bytes - register write mode

In register write mode, the command data is written to the special function register that is selected by the header of the write command.

The output of the internal registers controls various functions of the chip.

Similarly, in order to write to another register, the host needs to toggle SPI.SS and send another command header byte with the new register address.

3.4.6 Command data bytes – Start ADC command

There is no data byte associated with Start ADC command.

Chapter 4 Procedure to Perform Pixel Integration and Image Capture

This chapter outlines the steps to perform image capture with ULS24. Below is a list of basic steps to set up the device, perform integration and read out data.

1. Choose operation modes, and set operation parameters.
2. Initialize the device by performing write to a set of internal registers.
3. Send pixel control commands and perform integration and read out ADC data
4. Data conversion and processing

The ULS24 device has many operation modes to choose from. These operation modes are suitable for different application scenarios. The detailed relationship between operation modes and application scenario is beyond the scope of this document.

4.1 ULS24 operation modes

4.1.1 Image format

4.1.1.1. *Single row vs. Frame capture*

It is possible to capture image with a single row of pixels. We can either capture a 12X1 image or 24X1 image in this mode.

4.1.1.2. *Frame capture*

When we capture image with all rows of pixels in the ULS24 device, we have capture a frame. This can be done in 12X12 or 24X24 mode.

4.1.2 Image resolution

As described in the last chapter, the ULS24 device has 12X12 pixel groups. Each group consists of 4 adjacent pixels.

In 12X12 mode, we capture a frame consists of 12X12 pixel groups. Each pixel group may contain active 1~4 pixels. The choice of pixels are determined by the “binning pattern”, which the host controller can choose.

In 24x24 mode, we performed 4 12X12 image capture in a sequence. Each time, we change the binning pattern so that all four individual pixels are used one by one to perform integration.

4.1.3 Gain mode

Each pixel in the imager has operation modes which will affect its sensitivity and dynamic range. In the high gain mode, the pixels have higher light sensitivity than in the low gain mode. The responsivity ratio between high gain mode and low gain mode is 8 to 1.

4.1.4 Extended dynamic range mode

We can extend the dynamic range of the pixels by performing multiple image capture in different gain mode. Details of this technique is covered in “Appendix D Extend the dynamic range of ULS24 by performing multiple integrations in image capture”.

4.1.5 Integration time

Integration time is the amount of time each pixel uses to collect light. It is not the same, but related to frame time, which is the amount of time the image sensor uses to collect light for the frame.

Frame time is longer than the integration time because of command overhead (time to transmit commands) and data transfer time.

The host controller controls the amount of time delay between pixel reset and pixel read out. This delay in time becomes the integration time for the pixels receiving the commands. Thus, it is the host controller, not the ULS24 device, that provides the timing base for determining integration time.

4.1.6 Pixel binning pattern: 4-bin mode

In 12x12 mode, each pixel group consists of 4 small pixels. Any combination of the 4 small pixels can be turned on or off by software. This is accomplished by configuring the 4-bin mode input parameter. 4-bin mode parameter is a 4 bit binary entity. Appendix A explains how to bin small pixels to a pixel group.

In 24x24 resolution, the binning mode should be controlled automatically to cycle through all 4 pixels in a pixel group.

4.1.7 Other operation parameters

There are many other operation parameters that needs to be set by the host controller. The setting of some of these parameters are defined in the “trim data” file provided by the factory. For details of how to set these parameters, please see Appendix C.

4.2 Initializing the ULS24 device by sending write command to the special function registers (SFRs)

These steps are needed when we power up the ULS24 chip. When the ULS24 goes into power down mode and restart, this sequence needs to be performed again.

Every time we change the operation modes of the device, we need to perform this initialization sequence again.

Each special function register write command sequence consists of two bytes of SPI data. We start a command sequence by pulling SPI SS signal from high to low. We end a command sequence by pulling SPI SS signal from low to high.

4.2.1 Step 1: Program RampGen Trim Command sequence

Command header

Field	Bit	Value
Command	7:0	0x45

Command data

Field	Bit	Value
RampGen Trim	7:0	RampGen Trim Val

“RampGen Trim Val” is the actual value to be programmed to RampGen register. This value should be obtained from the trim data file. See Appendix C for more information about the trim data file.

4.2.2 Step 2: Program and Range Trim Command sequence

Command header

Field	Bit	Value
Command	7:0	0x44

Command data

Field	Bit	Value
Range Trim	7:0	0Fh

4.2.3 Step 3: Program the gain switch and binning mode

Command header

Field	Bit	Value
Command	7:0	0x41

Command data

Field	Bit	Value
Gain mode switch	7:4	0x0: High gain 0x1: Low gain
Binning pattern	3:0	Binning pattern value from 0 ~ 0xF. see Appendix for further explanation

4.2.4 Step 4: Program V20 and V15 Trim

As part of the initialization procedure, we need to program the V20 register according to the values provided by the trim data file. The V20 value is different for low gain and high gain mode.

We should program the V15 value to its default value of 0x8.

Command header

Field	Bit	Value
Command	7:0	0x42

Command data

Field	Bit	Value

V20 Trim	7:4	V20 Trim Value
V15 Trim	3:0	0x8

“V20 Trim Val” is the actual value to be programmed to V20 register. This value should be obtained from the trim data file. See Appendix C.

4.3 Perform pixel integration by sending pixel driving commands

Below we describe first the steps to perform pixel control and read out for a single row of pixel groups.

There are several actions performed to a row of pixels: reset, integration, read out, ADC convert, and data transfer.

4.3.1 How to set integration time

Integration for the host controller is merely a time delay between the end of reset sequence and beginning of read out sequence. The host controller is entirely responsible for controlling the length of the delay time. A reasonable approach is for the host controller to use its internal timer to time the delay and generate an interrupt to trigger the read out sequence. The ULS24 does not have an internal timer of its own.

Also note read out is to allow internal circuitry to read out pixel data and feed the data to ADCs for analog to digital conversion. After ADC conversion is done, the host control can then retrieve the data from data buffer with SPI read, and this sequence is called "data transfer".

4.3.2 Reset command sequence

The reset is performed on a given row of pixel groups by sending the following command sequence of 5 command bytes. The first command byte specifies the command type (pixel driving) and row number.

Command header

Field	Bit	Value
Command type	7:4	0x6
Row Number	3:0	0x0 to 0xB for Row 1 to Row 12

Command data

Field	Bit	Value
Command Byte 1	7:0	0xC0
Command Byte 2	7:0	0xE0
Command Byte 3	7:0	0x60

Command Byte 4	7:0	0x0

4.3.3 Read out sequence

The read out is performed on a given row of pixel groups by sending the following command sequence of 8 command bytes. The first command byte specifies the command type (pixel driving) and row number.

Command header

Field	Bit	Value
Command type	7:4	0x6
Row Number	3:0	0x0 to 0xB for Row 1 to Row 12

Command data

Field	Bit	Value
Command Byte 1	7:0	0xA1
Command Byte 2	7:0	0xB5
Command Byte 3	7:0	0x21
Command Byte 4	7:0	0x61
Command Byte 5	7:0	0x29
Command Byte 6	7:0	0x21
Command Byte 7	7:0	0x0

4.3.4 ADC conversion

After read out, the host should issue an ADC conversion command sequence. The ADC conversion command is only one byte.

Command header

Field	Bit	Value
ADC Conversion Command	7:0	0x80

Note there is no need to specify a row number, as there is only one array (12) of ADC converters, shared by all the rows of pixel groups.

ULS24 had an output pin called ADC Rdy. It is normally high. When ADC conversion process starts, it goes low. It takes approximately 83us to perform ADC conversion. After the ADC conversion process is finished, ADC Rdy signal will go from low to high.

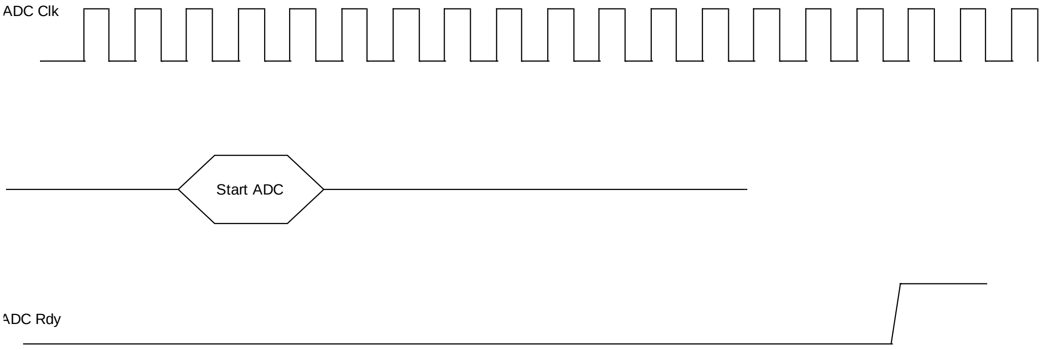


Figure 4.1 ADC Timing sequence

The host controller should monitor the ADC Rdy pin while waiting for ADC Conversion to finish. When a low to high transition of ADC Rdy pin has occurred, the host controller can then start perform data transfer.

4.3.5 Data transfer process

As described in Chapter 3, section “The host-transmit-ULS24 - receive (write) path”, the host performs data transfer be pulling SPI.SS signal from high to low, and issue a sequence of 24 bytes read.

After the data transfer process, the host controller has received 12 16-bit values from the row of 12 pixel groups. So for each pixel group, a 16-bit value is available. This 16 bit value needs to be converted to a final 12 bit pixel data output value, using an algorithm provided by Anitoa. The algorithm is embedded in a C code explained in “Appendix C. Trim Data File and TrimReader Code”.

4.3.6 Single row vs whole frame image capture

Thus the host can use the SPI interface to control the pixel reset, integration, read out, perform analog to digital conversion (ADC), and transfer the pixel data out.

Note that the operations described above are to be performed to a row of pixel group at a time. The timing is shown below:

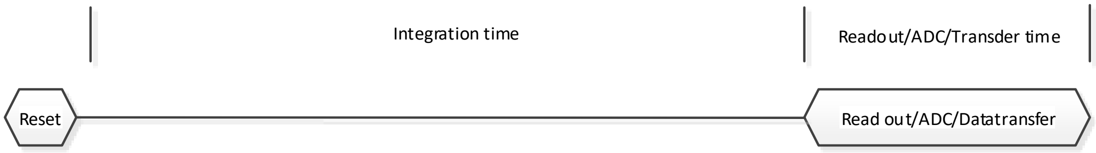


Figure 4.2 Pixel integration, sample, ADC and data transfer timing

The integration time is chosen by the host controller to achieve the desired sensitivity and dynamic range.

The read out, ADC, and data transfer time for the entire row of pixel group is about 350us. This is assuming 1MHz of SPI Clock rate.

©2014-2016, Anitoa Systems, LLC
Draft

4.3.7 Timing for read out the entire pixel array

4.3.7.1. *Non- pipelined mode*

Now, in order to perform integration, readout and ADC, data transfer for the entire 12 rows of pixel group, we can repeat the single row procedure 12 times.

4.3.7.2. *Pipelined mode*

To reduce frame time, we can use pipelined sequence to performed capture for multi rows of pixel groups as illustrated below.

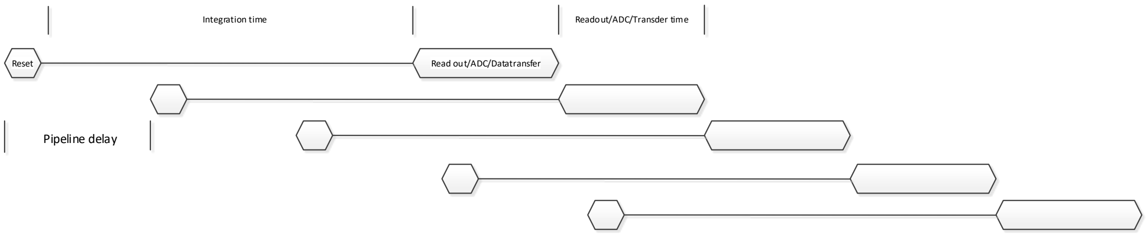


Figure 4.3 Pixel integration, sample, ADC and data transfer timing for multiple rows in pipelined mode

So the pixel integration time can overlap. However, the sample and ADC, and data transfer process for each row of pixels would have to be performed sequentially with no overlap time. Since the pixels that belong to different rows in a given column shares the same ADC.

Thus to make this work, the pipeline delay and integration time needs to satisfy the following requirements.

- Pipeline delay longer than Readout/ADC/Transfer time (approx. 350us)
- Integration time needs to be longer than the pipeline delay

Draft

4.3.8 Capture Image in 24x24 mode

To capture image in 24x24 mode, we just perform a sequence of 4 12X12 capture, with the binning pattern set to 0x1, 0x2, 0x4 and 0x8. The final data are then stitched to form a 24x24 image.

©2014-2016, Anitoa Systems, LLC

Chapter 5 Special Function Registers

5.1 Special function register map

ULS24 contains several special function registers whose output controls the internal circuitry. These are:

5.1.1.1. *Address: 0x01: [PIX_CTRL] – Pixel Control Register*

Field Name	Bit	Type	Default	Description
RESERVED	7:5	RO	0x0	Reserved
[SWITCH]	4	RW	0x0	Switch: this bit selects the gain mode of all pixels 0: High gain mode 1: Low gain mode
[BIN]	3:0	RW	0x0	Binning patter: this field sets the binning pattern for all pixel groups. Allowable values are 0 to 0xF. See Appendix A for details about pixel binning pattern.

5.1.1.2. *Address: 0x02: [OFST_TRIM] – Offset Trim Register*

Field Name	Bit	Type	Default	Description
[V20_TRIM]	7:4	RW	0x8	V20 Trim Value
[V15_TRIM]	3:0	RW	0x8	V15 Trim Value

5.1.1.3. *Address: 0x03: [PIX_TRIM] – Pixel Trim Register*

Field Name	Bit	Type	Default	Description
[IPIX_TRIM]	7:4	RW	0x8	Ipix Trim Value
[V24_TRIM]	3:0	RW	0x8	V24 Trim Value

5.1.1.4. *Address: 0x04: [RANGE_TRIM] – Range Trim Register*

Field Name	Bit	Type	Default	Description
RESERVED	7:4	RO	0x0	
[RANGE_TRIM]	3:0	RW	0x8	Range Trim Value

©2014-2016, Anitoa Systems, LLC

Draft

5.1.1.5. *Address: 0x05: [RAMPGEN_TRIM] – Rampgen Trim Register*

Field Name	Bit	Type	Default	Description
[RANGE_TRIM]	7:0	RW	0x88	Rampgen Trim Value

5.1.1.6. *Address: 0x06: [AMUX_SEL] – Amux Select Register*

Field Name	Bit	Type	Default	Description
RESERVED	7:3	RO	0x0	
[AMUX_SEL]	2:0	RW	0x0	Analog Mux Output Select

Many of these registered are called trim registers. Their purpose is to allow fine adjustment be made to the internal circuitry of the chip as a means to adjust the operating conditions. This capability is useful for manual or automatic tuning and calibration.

5.1.2 Further explanation of Special Function Register fields

5.1.2.1. *Rampgen trim*

This 8-bit register allows the host to adjust the slope of the internal ramp generator for the ADCs.

5.1.2.2. *Range trim*[6]

This 4-bit trimming value allows the possibility of adjusting the input dynamic range of the ADCs. The ADC input dynamic range is normally 2V. This register allows about +-15% adjustment of the range. The dynamic range of ADC can be monitored by measuring Vcm[7].

The ADC input dynamic range ~= Vcm* 1.6. Vcm can be monitored through the Amux output. The ADC input low boundary is 0.6 * Vcm, the ADC input high boundary is 1.4 * Vcm.

The nominal value of Vcm is 1.25V. The nominal ADC dynamic range is 2.0V.

Range trimming needs to be performed before Ramgen slope trimming.

5.1.2.3. *V24 and Ipix trimming*

V24 and Ipix are related to the operations of the pixels. Their value can be trimmed to achieve the best pixel performance.

5.1.2.4. *Switch*

The pixels of ULS24 have a low gain mode (low sensitivity) and high gain mode (high sensitivity). This flexibility allows us to achieve better pixel dynamic range. The switch bit is used to select low gain vs high gain mode.

'0' selects high gain. '1' selects low gain.

[6] The user does not need to care about this level of detail. Just set Range trim to 0xf is sufficient.

[7] Vcm is the common mode voltage of the buffer amplifier in the ADC.

5.1.2.5. *Tx binning pattern*

As mentioned before. The ULS24 contains 12 x 12 pixel groups, each of which contains 2x2 pixels.

At a given time, only pixel group can be selected to perform integration and read out. Within the pixel group, any combination of the regular pixels can be selected to contribute to light sensing. To achieve maximum resolution, the 4 regular pixels are selected one at a time to perform sensing; and 4 total passes of the entire 12 x 12 pixel group array needs to be performed to allow all pixels to sense light independently.

To achieve maximum sensitivity as the cost of reduced resolution, it is also possible to select all 4 pixels within a pixel group to sense light and perform one pass through the 12 x 12 pixel group array.

Any other combination of pixels within a pixel group is allowed.

5.1.2.6. *Amux select*

Certain internal analog voltage levels of ULS24 can be monitored from the pin for testing and calibration purposes. These signals are connected to an analog mux. A 3-bit digital input is fed to the Amux to select which signal is sent to the output of the Amux to be monitored from outside.

Chapter 6 The Temperature Sensor

6.1 Temperature sensor

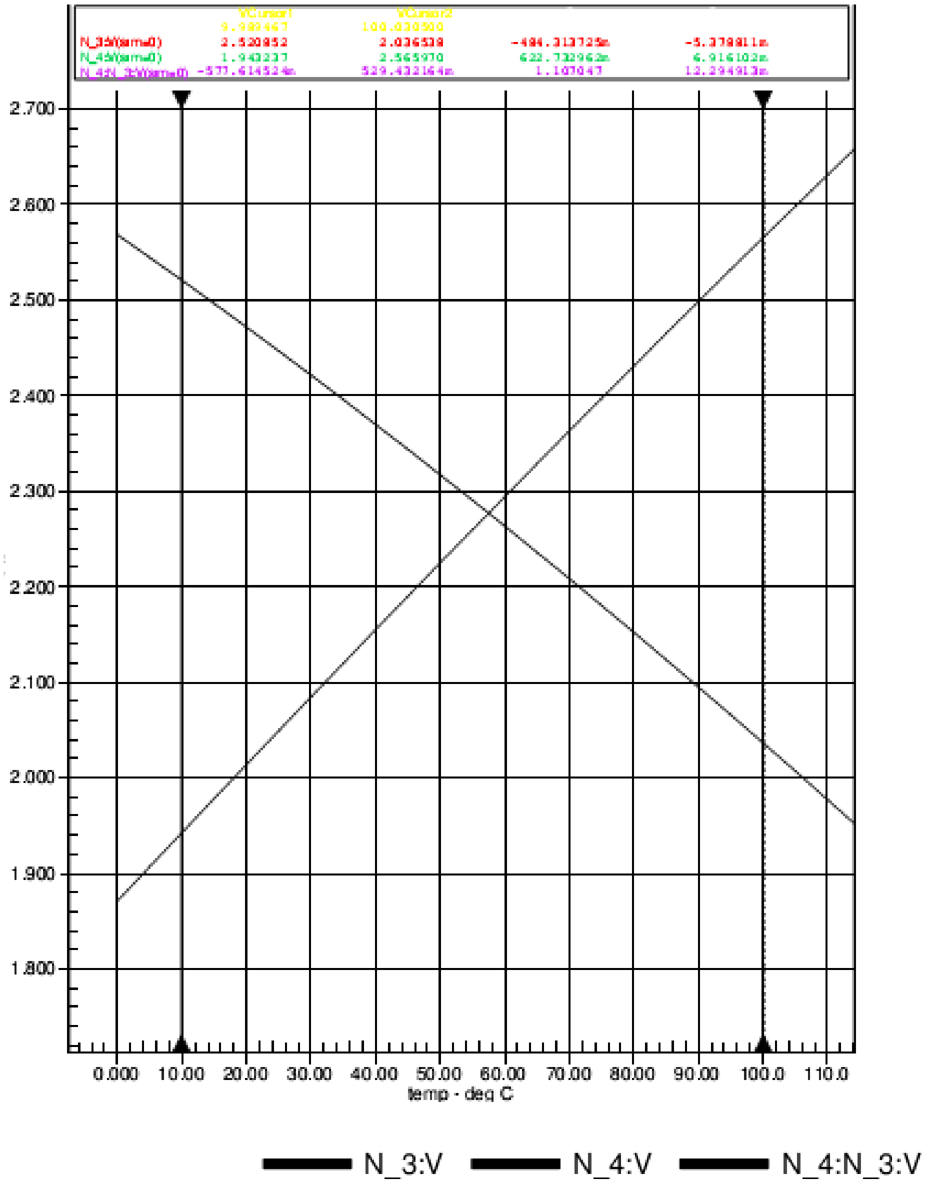
The ULS24 contains a temperature sensor that generates a analog differential output. A pair of pins, Tempoutp and Tempoutm, are used as the temperature sensor output.

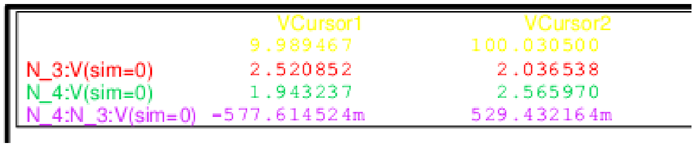
The voltage level of Tempoutp increases with the increase of temperature. The voltage level of Tempoutm decreases with the increase of the temperature. The two outputs cross at about 60 ºC.

Below is a reference design circuitry for temperature output processing, and relationship curve between Tempoutp and Tempoutm, and the temperature of the chip. The output of the reference circuitry should be fed to an ADC input in a microcontroller.

©2014-2016, Anitoa Systems, LLC

Draft





	VCursor1	VCursor2
	9.989467	100.030500
N_3:V(sim=0)	2.520852	2.036538
N_4:V(sim=0)	1.943237	2.565970
N_4:N_3:V(sim=0)	-577.614524m	529.432164m

©2014-2016, Anitoa Systems, LLC

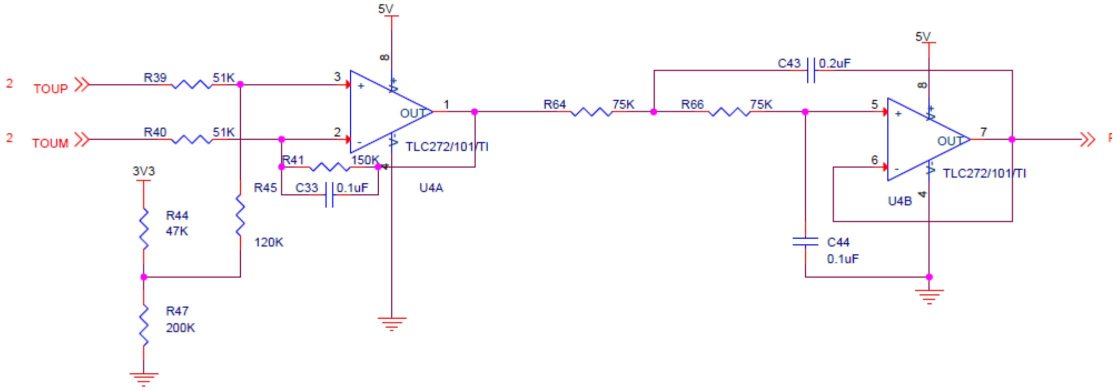


Figure 6.1. Temperature Sensor Simulation Data

Chapter 7 Appendix A ULS24 Pixel Format and Binning Pattern Explained

The orientation of the chip relative to the package is shown below.

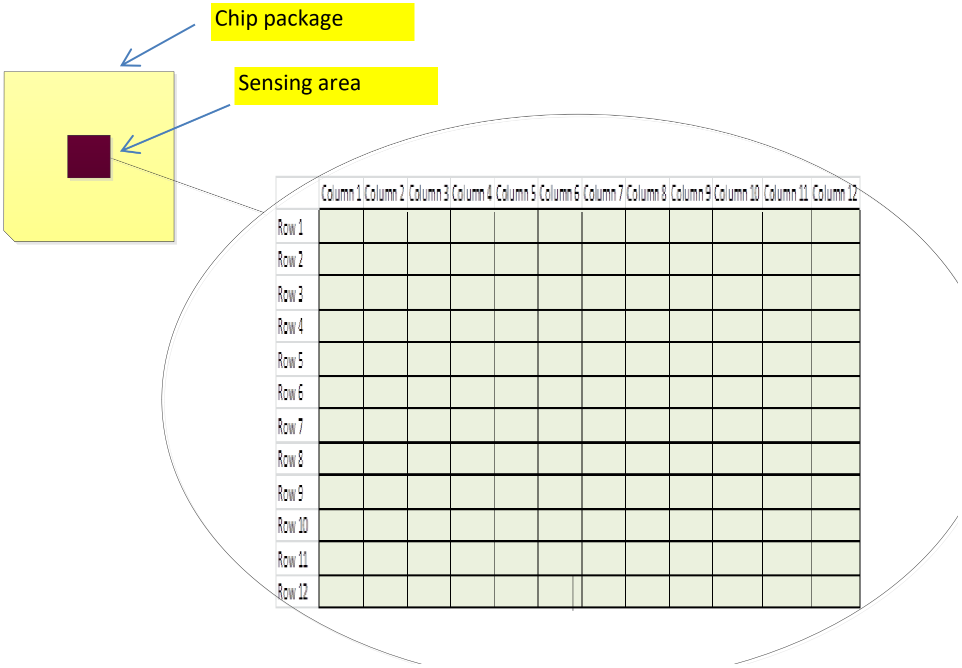


Figure 7.1 Chip orientation and pixel format

The pixels of the imager are arranges as a matrix shown below. Each pixel group is comprised of 4 subpixels.

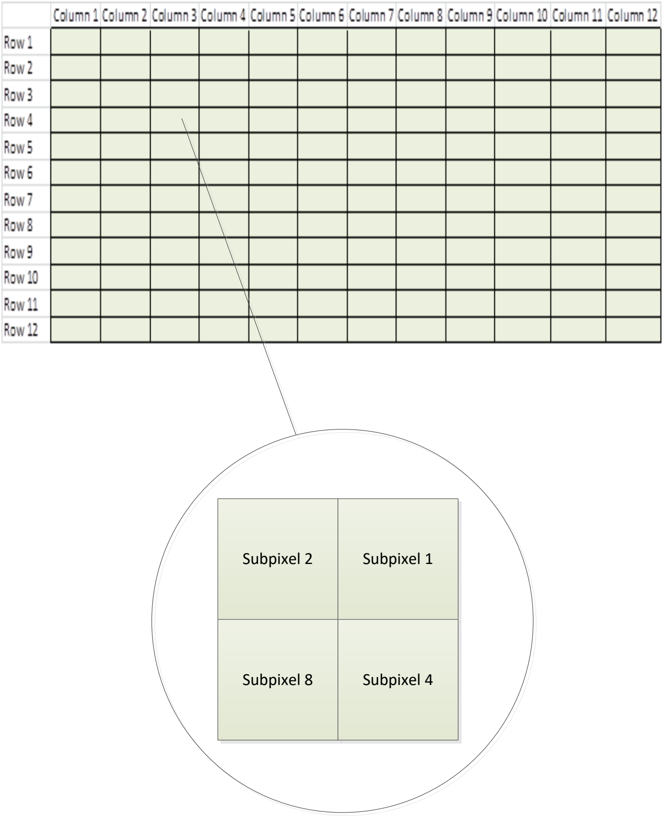
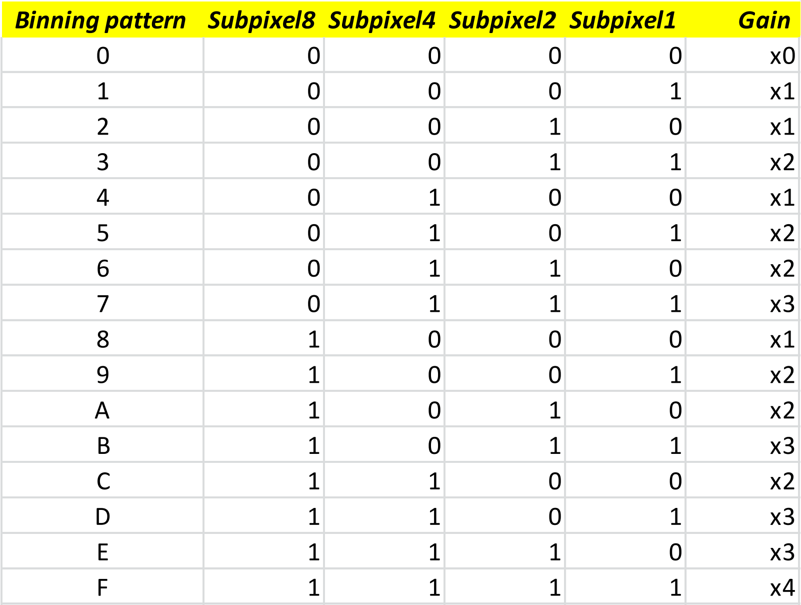


Figure 7.2, Pixel binning pattern

In 12X12 low resolution mode, each effective pixel is comprised of 4 subpixels. Any combination of the 4 subpixels can be chosen to sense light. When more than one subpixel is chosen, the sensitivity is multiplied. For example, if all 4 pixels are chosen, the sensitivity of the device is quadrupled.



Binning pattern	***Subpixel8***	***Subpixel4***	***Subpixel2***	***Subpixel1***	***Gain***
0	0	0	0	0	x0
1	0	0	0	1	x1
2	0	0	1	0	x1
3	0	0	1	1	x2
4	0	1	0	0	x1
5	0	1	0	1	x2
6	0	1	1	0	x2
7	0	1	1	1	x3
8	1	0	0	0	x1
9	1	0	0	1	x2
A	1	0	1	0	x2
B	1	0	1	1	x3
C	1	1	0	0	x2
D	1	1	0	1	x3
E	1	1	1	0	x3
F	1	1	1	1	x4

Figure 7.3 Pixel pinning pattern code explained (for column 2,3,4,5: ‘1’ means pixel selected; ‘0’ means pixel not selected).

Chapter 8 Appendix B Power Up Sequence and ADC Timing

8.1 Power up sequence

The sequence below shows the start-up condition for ULS24

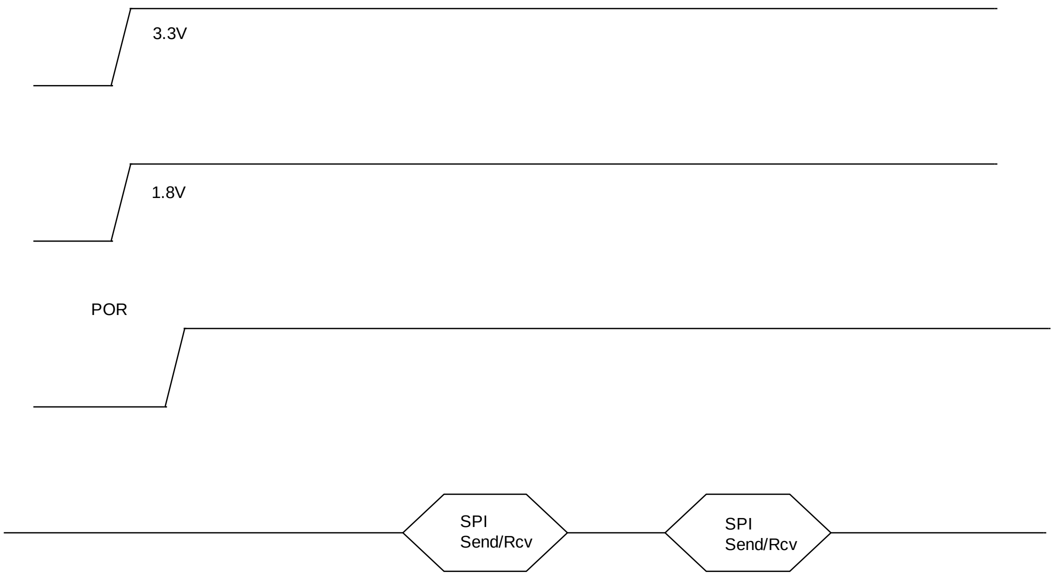


Figure 7.1 Chip start up timing

After the power supply, AVdd (3.3V) and VDD (1.8V) becomes stable, the power on reset, POR should release (change level from low to high). The chip is now ready. At this time, the host can communicate with ULS24 through SPI interface.

The POR should be released 5us or longer after the power supply becomes stable. The POR reset to power stable delay can be arbitrarily long.

During start up and the initial calibration phase, it is not necessary to provide ADCClk.

8.2 Timing for ADC

Before starting ADC, a 12MHz ADCClk needs to be provided to the ADCClk input. A SPI command can be sent to start ADC conversion. After Start ADC command is issued, the ADC Rdy signal goes low, indicating the ADC process has started.

When ADC conversion is finished, ADC Rdy signal will toggle from low to high. The host can use this as an indication that it can now read out the content of the data buffer registers.

ADCRdy will go low again when ADC starts again.

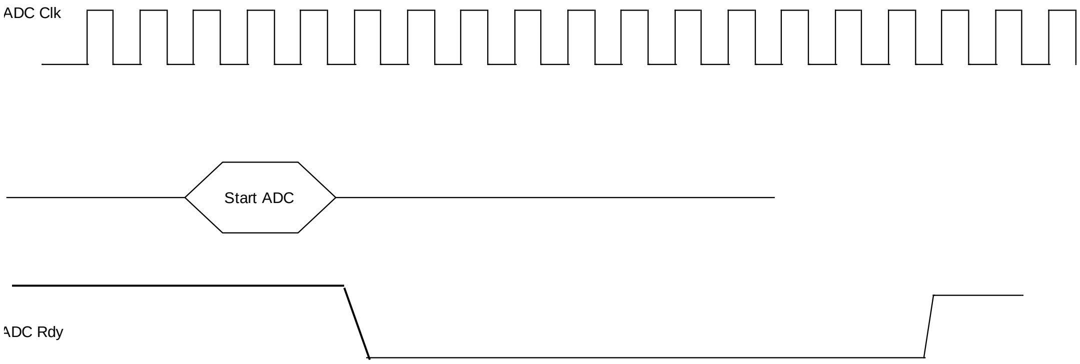


Figure 7.2 ADC timing

8.3 Power saving modes

Power saving is important and beneficial in at least two use case:

1. Applications where ULS24 is part of a system powered by battery, especially small battery in a miniaturized device.
2. Applications where ULS24 is not dissipating heat very well and used in low light sensing. It is desirable for ULS24 to keep cool and avoid temperature build up which will affect dark current.

There are two ways to save power for ULS24. The system can stop the clock when ADC is not operating (e.g. during integration time) and only start the clock when ADC is about to start. Since only the ADC needs the external 12MHz clock to operate, this method will save the switching power in ADC digital circuits.

It is also possible to cut all the power supply – analog 3.3V and digital 1.8v, as well as clocks to UYLS24, thus completely eliminate power. In this case, the system just need to apply the whole power up sequence again when start ULS24 into function mode.

Chapter 9 Appendix C. Trim Data File and TrimReader Code

For each ULS24 device, a trim data file with the file name of “trim.dat” is provided. An example is given here in this chapter. In addition, a companion trim data read out and ADC data

```
DEF 509 {

       Kb {
              -0.348717949 ,      40.74358974   ,
              -0.093877551 ,      14.46938776   ,
              -0.261538462 ,      28.30769231   ,
              -0.353846154 ,      35.76923077   ,
              0.194871795  ,      5.025641026   ,
              -0.020512821 ,      -58.8974359   ,
              -0.394871795 ,      35.97435897   ,
              -0.138461538 ,      75.69230769   ,
              -0.020512821 ,      -23.8974359   ,
              -0.425641026 ,      99.12820513   ,
              -0.179487179 ,      10.8974359    ,
              -0.512820513 ,      102.5641026
       }

       Fpn_lg {
              97.91666667,  90.13888889,  149.5972222,  84.41666667,  119.375,
              98.56944444,  103.5972222,  157.7777778,  93.97222222,
              164.5833333,  185.1111111,  205.0277778
       }

       Fpn_hg {
              114.6527778,  106.75,       166.625,      102.1944444,
              137.6805556,  117.0972222,  124.125,      177.0972222,
              110.6388889,  183.8055556,  204.2916667,  256.7638889
       }

       AutoV20_lg {
              0x09
       }

       AutoV20_hg {
              0x0b
       }

       Rampgen {
              0xa4
       }
```

processing C code – TrimReader.cpp is also provided. The user software can incorporate this code to read out the trim files and process ADC data read from the ULS24 device.

Some of the important parameters are Rampgen value and V20 value. These needs to be programmed to the chip in the initialization stage, see section "Initialize the device by performing write to a set of internal registers." Note that V20 value is different for low gain and high gain mode.

The other parameters in the trim file are used by TrimReader software to perform conversion of 16 bit ADC data to the final digital output for the pixels.

In the TrimReader module, there is a function called:

```
int ADCCorrection(int ColNum, BYTE HighByte, BYTE LowByte, int
GainMode);
```

The host software should pass the 16-bit pixel ADC data to "HighByte" and "LowByte" field, specify the Column number "ColNum", GainMode ('0' High gain, '1' Low gain). In return, the function will provide a 12 bit (0 to 4095) pixel digital output value.

Below is a list of the TrimReader header file

```
#define TRIM_IMAGER_SIZE 12

class CTrimNode {

public:

       double kb[TRIM_IMAGER_SIZE][2];
       double fpn[2][TRIM_IMAGER_SIZE];  // 0 - lg, 1 - hg
       unsigned int rampgen;
       unsigned int auto_v20[2];
       double tempcal[TRIM_IMAGER_SIZE];

       CString name;

public:

       CTrimNode();

};

#define TRIM_MAX_NODE 4
#define TRIM_MAX_WORD 640

class CTrimReader {

protected:

       CFile InFile;

public:

       CTrimNode Node[TRIM_MAX_NODE];
       CTrimNode *curNode;
       int NumNode;

public:

       CTrimReader();
       ~CTrimReader();

       int Load(TCHAR*);
       void Parse();
       void ParseNode();

       int GetNumNode() {
              return NumNode;
       }

       int ADCCorrection(int ColNum, BYTE HighByte, BYTE LowByte, int GainMode);

};
```

Below is an example of how to load trim.dat file with the TrimReader in the main program, for example, an initialization function in a Windows® program:

```
GetCurrentDirectory(MAX_PATH, g_CurrentDirectory);

CString path;
path = g_CurrentDirectory;
path += "\\Trim\\trim.dat";

LPTSTR lpszData = path.GetBuffer(path.GetLength());

int e = m_TrimReader.Load((TCHAR*)lpszData);

path.ReleaseBuffer(0);

if(e) {
       m_TrimReader.Parse();
}
```

Chapter 10 Appendix D Extend the dynamic range of ULS24 by performing multiple integrations in image capture

The ULS24 pixels are connected to 12 bit ADCs to produce a digital output ranging from 0 to 4095. This dictates the output dynamic range when a single image is captured in one integration period.

It is however possible to extend the output dynamic range of ULS24 by performing an image capture with multiple integration procedures. In order for this to work, several conditions are necessary:

1. The total time needed to capture image will increase
2. The light emission level from the sample is assumed to be constant during the time of integrations
3. The sensitivity ratio between the two integrations is known accurately.

The example below shows how to perform such extended dynamic range image capture with the combination of integration in “low gain” and “high gain” mode.

In this example, we perform two integrations back to back, using the same integration time. The first integration is performed in low gain mode; the second integration is performed in high gain mode. The sensitivity ratio between high gain mode and low gain mode is known to be 8. The output data is then calculated from the sample code provided below.

```
#define DARK_LEVEL                    0       // Numerical dark level
#define SATURATION_LEVEL_DIV_8        480     // Should be 512 theoretically, but using 
480 to give it some margin.
#define DATA_SIZE                     24      // Use 12 in 12X12 mode

int data_l[DATA_SIZE][DATA_SIZE];     // Low gain data
int data_h[DATA_SIZE][DATA_SIZE];     // High gain data

int data_o[DATA_SIZE][DATA_SIZE];     // Output data

void CalcHDR()
{
        int i, j;

        for(i=0; i<DATA_SIZE; i++) {
                for(j=0; j<DATA_SIZE; j++) {
                        if((data_l[i][j]) < SATURATION_LEVEL_DIV_8 + DARK_LEVEL) {
                                data_o[i][j] = data_h[i][j];
                        }
                        else {
                                data_o[i][j] = (data_l[i][j] - DARK_LEVEL) * 8 + 
DARK_LEVEL;
                        }
        }
}
```

It is also possible to extend the dynamic range of ULS24 by performing two integrations with the same gain mode (e.g. high gain mode), but different integration time. ULS24 has excellent response linearity with regarding to integration time. For example, the response of the pixels to a given level of light input at 8s integration time will be 8 times the response with 1s integration time. Because of this property, we expect to achieve good results with extending the dynamic range of ULS24 by performing multiple integrations with different integration time.

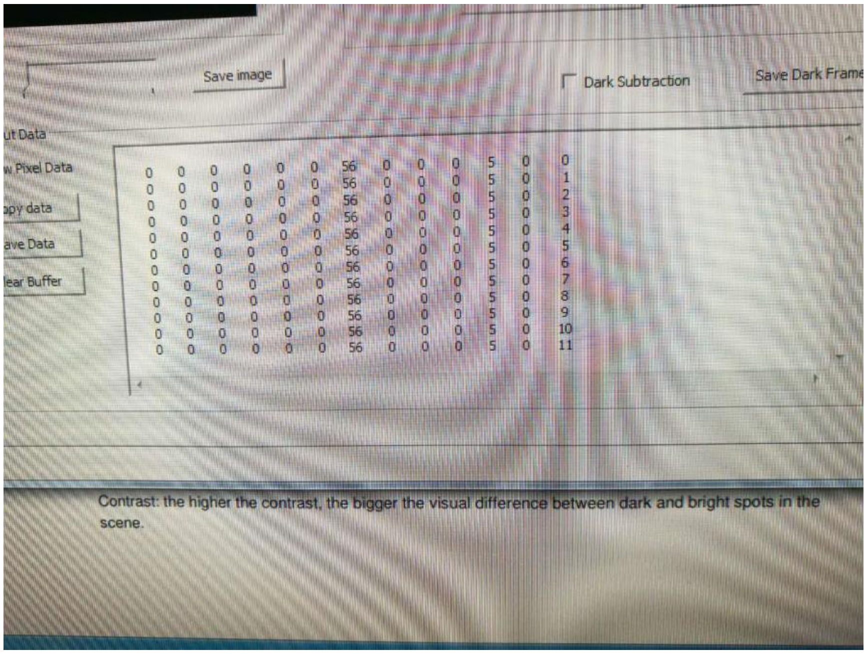
Chapter 11 Exposure saturation analysis

ULS24 pixels are highly sensitive. This means unless the ULS24 device is placed in a dark environment, the pixels will easily saturate.

Because of Erata #1, when ULS24 pixels saturate, instead of outputting value 0xFFF, the output will freeze (not update). In other words, when the host performs a read operation, the returned pixel exposure value will be whatever is left in the output register from the last frame.

One way to judge whether the pixels are in a saturated state is to observe that the pixel output values are identical for a given column.

For example, after power on reset, if we take an image while the sensor is exposed to strong ambient light, many of the pixel outputs maybe '0'. This is because the output register values were initialized to '0' and they are not updated due to saturation.



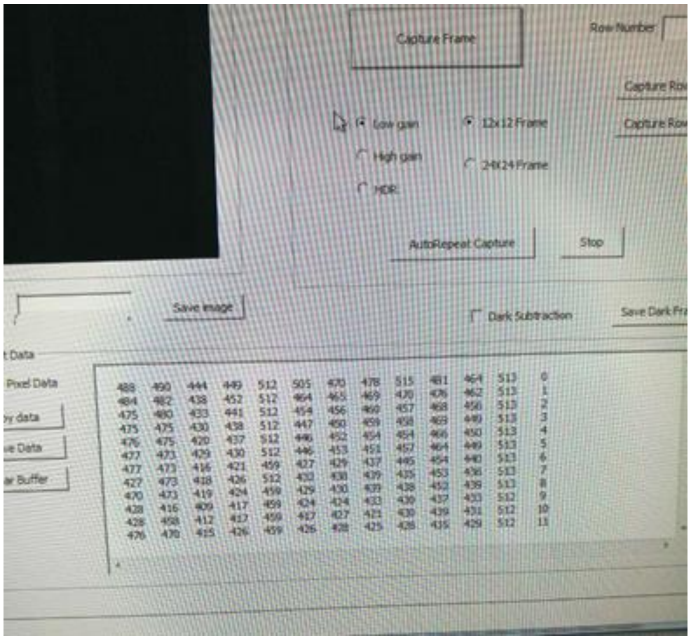
```
0  0  0  0  0  0  56  0  0  0  5  0  0
0  0  0  0  0  0  56  0  0  0  5  0  1
0  0  0  0  0  0  56  0  0  0  5  0  2
0  0  0  0  0  0  56  0  0  0  5  0  3
0  0  0  0  0  0  56  0  0  0  5  0  4
0  0  0  0  0  0  56  0  0  0  5  0  5
0  0  0  0  0  0  56  0  0  0  5  0  6
0  0  0  0  0  0  56  0  0  0  5  0  7
0  0  0  0  0  0  56  0  0  0  5  0  8
0  0  0  0  0  0  56  0  0  0  5  0  9
0  0  0  0  0  0  56  0  0  0  5  0  10
0  0  0  0  0  0  56  0  0  0  5  0  11
```

Contrast: the higher the contrast, the bigger the visual difference between dark and bright spots in the scene.

Figure 11.1 Picture saturation. Pixels output from a given column are exactly the same. This is the first frame after power up and the outputs are saturated.

©2014-2016, Anitoa Systems, LLC

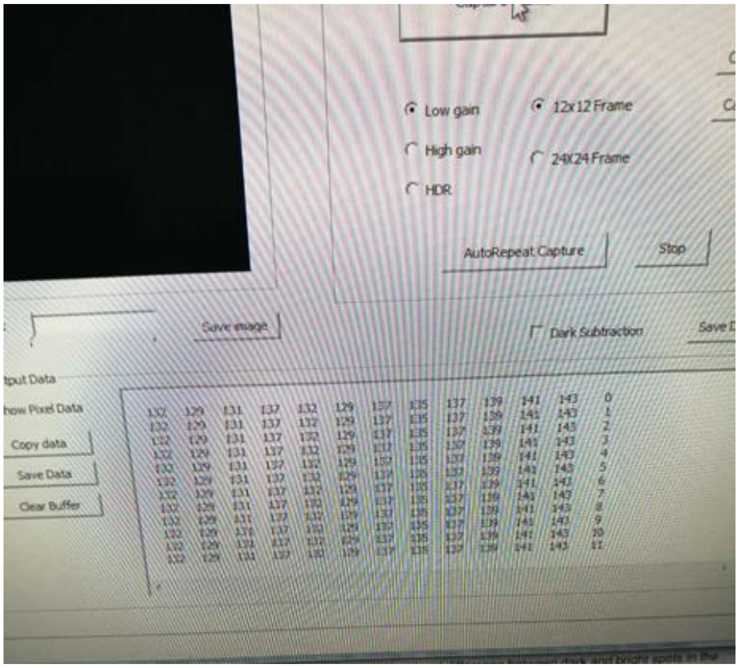
Now, if we place the sensor in a darker environment, we should see pixel output values being updated properly, as shown in the screenshot below.



```
488  490  444  449  512  505  470  478  515  481  464  513   0
484  482  438  452  512  464  465  469  470  476  462  513   1
475  480  433  441  512  454  456  460  457  468  456  513   2
475  475  430  438  512  447  450  459  458  469  449  513   3
476  475  420  437  512  446  452  454  454  466  450  513   4
477  473  429  430  512  446  453  451  457  464  449  513   5
477  473  416  421  459  427  429  437  445  454  440  513   6
427  473  418  426  512  432  438  439  435  453  436  513   7
470  473  419  424  459  429  430  439  438  452  439  513   8
428  416  409  417  459  424  424  433  439  437  433  512   9
428  458  412  417  459  417  427  421  430  439  431  512  10
476  470  415  426  459  426  428  425  428  435  429  512  11
```

Figure 11.2 Normal pixel output. Output values from a given column are different.

However, if we place the sensor back into a high ambient level environment, the pixel output will "freeze" again. This is seen by repeated pixel output value for each column. Although these values are no longer zero, due to the fact that the pixel output registers now contain non-zero values from previous image shots.



● Low gain
○ High gain
○ HDR

● 12x12 Frame
○ 24X24 Frame

□ Dark Subtraction

```
132  129  131  137  132  129  137  135  137  139  141  143   0
132  129  131  137  132  129  137  135  137  139  141  143   1
132  129  131  137  132  129  137  135  137  139  141  143   2
132  129  131  137  132  129  137  135  137  139  141  143   3
132  129  131  137  132  129  137  135  137  139  141  143   4
132  129  131  137  132  129  137  135  137  139  141  143   5
132  129  131  137  132  129  137  135  137  139  141  143   6
132  129  131  137  132  129  137  135  137  139  141  143   7
132  129  131  137  132  129  137  135  137  139  141  143   8
132  129  131  137  132  129  137  135  137  139  141  143   9
132  129  131  137  132  129  137  135  137  139  141  143  10
132  129  131  137  132  129  137  135  137  139  141  143  11
```

Figure 11.3 Saturation occurred again. Pixel outputs from a given column are identical.

©2014-2016, Anitoa Systems, LLC