## General Description

The epc660 is a fully integrated 3D-TOF imager with a resolution of 320 x 240 pixels (QVGA). It is a highly integrated system-on-chip camera system. Apart from the actual CCD pixel field, it includes the complete control logic to operate the device. Data communication is done through a high-speed digital 12-bit parallel video interface.

Even for mobile devices, only a few additional components are needed to integrate 3D camera capability. Depending on the system design, a resolution in the millimeter range for measurements up to 100 meters is feasible. 65 full frame TOF images are delivered in maximal configuration. By using the advanced operation modes, this can be boosted up to more than 1000 TOF images per second! The high degree of integration lays base for straight-forward camera system design with minimal part count. The extremely high sensitivity of the optical front end allows for a reduced illumination subsystem and reduces the power consumption of the overall system significantly.

An evaluation kit with hard- and software examples and a comprehensive manual helps the system designer to speed up system integration.

## Applications

- People detection and counting
- Mobile postal parcel size measurement
- Machine safety
- Helicopter near terrain flight assistance
- Car collision avoidance systems
- Pedestrian detection and breaking systems
- Man-Machine interface
- Gesture control
- Body size measurement
- General volumetric mapping

## Block Diagram

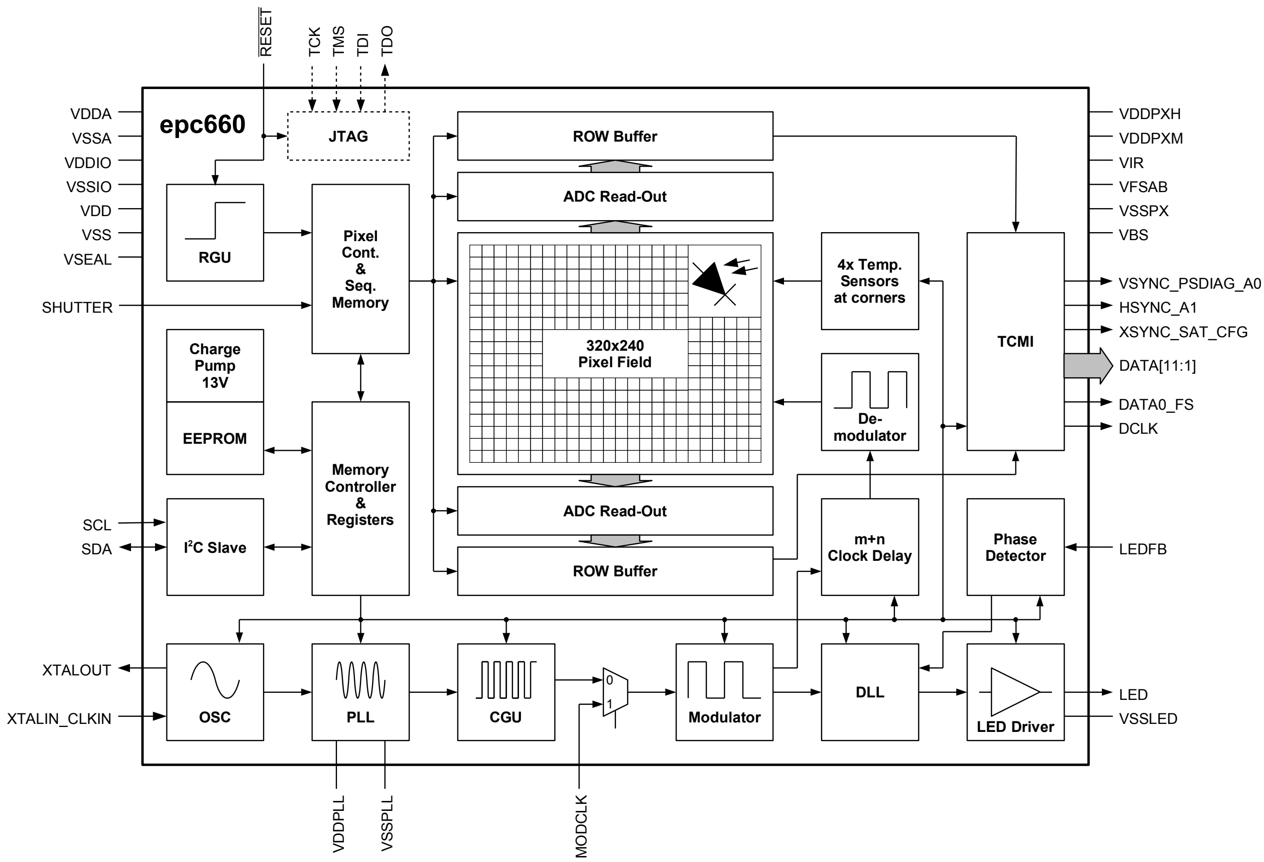



Figure 1: Functional Block Diagram

## Main Features

■ **General**
- □ 3D TOF imager in full monolithic design
- □ 320 x 240 pixel-field, backside illuminated
- □ 65.5fps full 3D TOF frame rate, single frame rate up to 262fps
- □ Region of interest setting allows up to several kfps
- □ Distance range up to 240 meters depending on system design
- □ 4 integrated temperature sensors for drift compensation

■ **Measurement performance**
- □ Absolute accuracy in the sub-centimeter range with appropriate setup and calibration

■ **Integrated LED (or laser diode) driver**
- □ LED feedback for drift compensation
- □ Laser diode (LD) illumination possible
- □ Single LED output pad, up to 350mA drive

■ **Parallel digital data interface TCMI**
- □ 80MS/s max. data rate, 2.5/3.3V compatible
- □ 12-bit parallel DATA output + XSYNC/SAT flag
- □ VSYNC, HSYNC ('ITU-R656 like' HW synchronization) and DCLK outputs

■ **I²C control interface (slave)**
- □ 400kHz (FM) / 1MHz (FM+)

■ **Integrated EEPROM 128 x 8-bit**
- □ Calibration data and user programmable parameters
- □ Unique chip ID

■ **System / Modulation clock**
- □ System clock 4MHz, internal by using crystal/resonator or using external input
- □ External LED/LD modulation input MODCLK (optional) up to 80MHz

■ **Power supply**
- □ Supply voltages +10V, +5V, +2.5/3.3V, +1.8V, -10V
- □ Power consumption approx. 0.75W (average)

■ **Packaging**
- □ 9.7x8.7mm cost optimized 68pin CSP (chip scale package),
- □ Backside illuminated flip-chip SMD mounting

■ **Other data**
- □ -40 ... +85°C operating range
- □ ROHS compatible

## Measurement Modes

■ **LED/LD modulation modes**
- □ Sinusoidal modulation for single camera applications
- □ PN modulation with programmable sequence lengths for multi-camera applications
- □ Selectable modulation frequencies 0.625 ... 20MHz resulting in unambiguity distance of 7.5m … 240m

■ **Distance measurement modes**
- □ 65.5 fps 3D TOF with 4x DCS frames, full pixel-field
- □ 131 fps 3D TOF with 2x DCS frames, full pixel field
- □ 262 fps 3D TOF with rolling read-out 4x DCS frames, full pixel-field
- □ Ultra fast measurement by reduction of the image field (ROI)
- □ SHUTTER release input for precise start/stop and single/continuous measurement control

■ **Non distance measurement modes**
- □ Ambient-light measurement (Black & white image without illumination)
- □ Black & white image with active illumination

## Readout Modes

■ **ROI (Region of Interest)**
- □ Rectangular sub-pixel-field read-out
- □ Increased frame rate

■ **Binning and resolution reduction**
- □ Binning of two adjacent pixels
- □ Resolution reduction by $n^{th}$ row or column, restricted to $2^{nd}$, $4^{th}$ or $8^{th}$ row or column (can be set independently) to read-out
- □ Increased frame rate for reduced number of rows

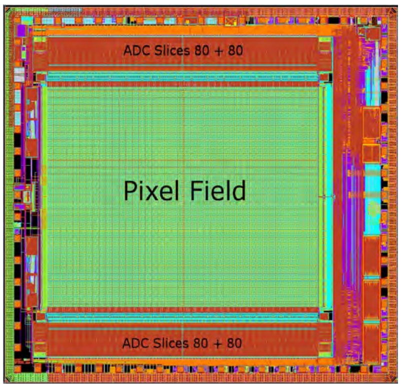


Figure 2: Picture of the epc660

# Table of Contents

# 1. Electrical, optical and timing characteristics

## 1.1. Operating conditions and electrical characteristics

Typ. operational ratings, $T_A$ = +25ºC, unless otherwise stated

| Parameter | Description | Conditions/Comments | $V_{SC}$ | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $V_{DD}$,$V_{DDPLL}$ | Digital supply voltage | Ripple [1] < ± 20 mV | $V_{DD}$ | 1.71 | 1.80 | 1.98 | V |
| $V_{DDIO}$ | IO supply voltage [3] | Ripple [1] < ± 50 mV | $V_{DDIO}$ | 2.25 | 2.5/3.3 | 3.63 | V |
| $V_{DDA}$, $V_{DDPXM}$, $V_{IR}$ | Analog supply voltage [2] | Ripple [1] < ± 20 mV | $V_{DDA}$ | 4.5 | 5.0 | 5.5 | V |
| $V_{DDPXH}$ | Pixel analog 2 supply voltage [2] | Ripple [1] < ± 20 mV | $V_{DDPXH}$ | 9.5 | 10 | 10.5 | V |
| $V_{BS}$ | Bias supply voltage | Ripple [1] < ±50 mV | $V_{BS}$ | -10.5 | -10.0 | -9.75 | V |
| $I_{VDD}$ | Digital supply current | @nominal voltage | | | 18 | | mA |
| $I_{VDDPLL}$ | PLL supply current | @nominal voltage | | | 4 | | mA |
| $I_{VDDIO}$ | IO supply current [4] | DCLK = 40MHz | | | 8 | | mA |
| $I_{VDDA}$ | Analog supply current | @nominal voltage | | | 125 | 350 | mA |
| $I_{VDDPXM}$ | Pixel analog 1 supply current | @nominal voltage | | | 1 | | mA |
| $I_{VIR}$ | Isolation supply current | @nominal voltage | | | 1 | | mA |
| $I_{VDDPXH}$ | Pixel analog 2 supply current | @nominal voltage | | | 13 | | mA |
| $I_{VBS}$ | Bias supply current [8] | Depending on the amount of light on the pixel field | | | 3.8 [8] | | mA |
| $V_{DDLED}$ | LED and LEDFB voltage range | | $V_{DDLED}$ | | | $V_{DDA}$ | V |
| $V_{LED\_OFF}$ | Off-voltage at output pin LED | | | | | $V_{DDLED}$ | V |
| $V_{LED\_ON}$ | On-voltage at output pin LED (forward voltage) | @ $I_{LED}$ = 200 mA | | | 200 | | mV |
| $I_{LED}$ | LED output sink current; on state | modulated peak current<br>continuous DC current | | | | 350<br>175 | mA<br>mA |
| $I_{LED\_LEAK}$ | LED output leakage current; off state | $V_{LED}$ max. | | | | 10 | µA |
| $V_{LEDFB}$ | LEDFB input voltage range | | $V_{DDLED}$ | 0 | | 8.5 | V |
| $V_{LEDFB\_P2P}$ | LEDFB input voltage peak to peak | | | 0.5 | | 1.8 | V |
| $V_{IH}$ | Digital high level input voltage [5] | | | 0.7 x $V_{DDIO}$ | | | V |
| $V_{IL}$ | Digital low level input voltage [5] | | | | | 0.3 x $V_{DDIO}$ | V |
| $V_{OH}$ | Digital high level output voltage [5, 6] | | | 0.8 x $V_{DDIO}$ | | | V |
| $V_{OL}$ | Digital low level output voltage [5, 6] | | | | | 0.2 x $V_{DDIO}$ | V |
| $I_{IH}$ | Digital high level input current [7] | $V_{IH}$ max. | | | | 10 [7] | µA |
| $I_{IL}$ | Digital low level input current [7] | $V_{IL}$ min. | | -10 [7] | | | µA |
| $I_{OH}$ | Digital output source current [7] | $V_{OH}$ max. | | | | 50 | mA |
| $I_{OL}$ | Digital output sink current [3] | $V_{OL}$ min. | | -50 | | | mA |
| $C_{IO}$ | IO load capacitance [5] | | | | | 30 | pF |
| $f_{IO}$ | IO switching frequency [5] | | | | 40 | 80 | MHz |
| $P_{Pk}$ | Power dissipation (average) | See Table 20 | | | 0.75 | | W |
| $R_{Th}$ | Thermal resistance | on PCB with underfill | | | | 40 | °K/W |
| $T_A$ | Operating temperature | | | -40 | | 85 | °C |

Table 1: Operating conditions and electrical characteristics

Notes:

1. Min. and Max. voltage values include noise and ripple voltages.
2. Analog voltage supplies have direct influence on measurement performance. They must be properly decoupled for low noise and ripple.
3. IO voltage supply must be fixed with respect to external processor's IO supply voltage levels used in the application. It can be set to any value within min and max. operating voltage.
4. When device is operated at max $f_{DCS}$ frame rate, DCLK at 40MHz, driving loads 15pF each.
5. Except $I^2C$ pins SCL and SDA. $I^2C$ pins SCL SDA are according to $I^2C$ standards.
6. $V_{OH/OL}$ and $I_{OH/OL}$ values are measured at max $C_{IO}$ and max $f_{IO}$.
7. Value is without termination resistors
8. A bright illuminated white target right in front of the chip with lens leads to an $I_{VBS}$ of approx. 3.8 mA. At room temperature, $I_{VBS}$ is approx. 3.6 mA without any illumination. $I_{VBS}$ with strong illumination (approx. 55 mW/cm$^2$, no lens) typ. 17 mA.

### 1.2. Absolute maximum ratings

| Parameter | Conditions |
|---|---|
| Supply voltage $V_{DD}$, $V_{DDPLL}$ | -0.5V ... +2.0V |
| Supply voltage $V_{DDIO}$, $V_{DDA}$, $V_{DDPXM}$, $V_{IR}$, $V_{DDLED}$ | -0.5V ... +5.5V |
| Supply voltage $V_{DDPXH}$ | -0.5V ... +13.5V |
| Supply voltage $V_{BS}$ | -12.0 ... +0.5V |
| Voltage to any pin in the same $V_{SC}$ supply class.<br>For Supply Classes refer to Table 1: Operating conditions and electrical characteristics.<br>For Supply Classes vs. pin correspondence refer Table 8: Pin list. | $V_{SC\ min}$ - 0.3V … $V_{SC\ max}$ + 0.3V |
| ESD rating<br>Note: This is a highly sensitive CMOS mixed signal device. Handling and assembly of this device should only be done at ESD protected workstations. | JEDEC HBM class 1C (1kV to < 2kV) |
| Storage temperature range ($T_S$) | -40°C to +85°C |
| Relative humidity | 0 ... 95%, non-condensing |

Table 2: Absolute maximum ratings

### 1.3. Temperature sensor characteristics

Typ. operational ratings, unless otherwise stated

| Parameter | Description | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $T_{TEMP}$ | Measurement range | | -40 | | +85 | ºC |
| $P_{TEMP}$ | Sensor resolution | | | 14 | | bit |
| k | Temperature sensor gain | | | 0.067 | | ºK/LSB |

Table 3: Temperature sensor characteristics

### 1.4. Timing parameters

Typ. operational ratings, $T_A$ = +25ºC, unless otherwise stated

| Parameter | Description | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $t_{STARTUP}$ | Start-up time | after applying external supplies | | 340 | 1'000 | µs |
| $t_{\overline{RESET}}$ | $\overline{\text{RESET}}$ | | 100 | | | ns |
| $t_{PLLStrap\_scan}$ | | | | 4x osc_clk | | |
| $t_{PLL}$ | PLL lock time | | | | 30 | µs |
| $t_{EEPROM\_to\_CFG}$ | Load CFG registers | Copy EEPROM to CFG registers | | 340 | | µs |
| $f_{XTAL}$ | Clock frequency | of the crystal oscillator (or ceramic resonator) | --- | 4 | --- | MHz |
| $df_{XTAL}$ | Clock frequency deviation | any deviation is added as a linear distance error | | | ±100 | ppm |
| $f_{JITTER}$ | Clock frequency phase jitter | peak-to-peak, cycle to cycle | | | 50 | ps |
| $f_{LED}$ | LED modulation frequency | | 0.625 | | 40 | MHz |
| $f_{MODCLK}$ | Ext. modulation clock | Refer to chapter 7.9. | | | 80 | MHz |
| $t_{LED\_rise/fall}$ | Required rise/fall time<br>of the illumination LED/LD | | | | 12 | ns |
| $f_{DCLK}$ | TCMI pixel rate | 12 bit pixel data + saturation flag | | 40 | 80 | MPixel/s |
| $f_{TCMI\_data}$ | TCMI data rate | | | 520 | 1'040 | Mbit/s |
| $f_{SCL}$ | $I^2C$ data rate | | | | 1 | Mbit/s |

Table 4: Timing parameters

### 1.5. Optical characteristics

Typ. operational ratings, $T_A$ = +25ºC, unless otherwise stated

| Parameter | Description | Conditions/Comments | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $A_{PIXEL}$ | Photosensitive area | 1 pixel | | 20 x 20 | | µm |
| $A_{SENSOR}$ | Photosensitive area | 320 x 240 pixel | | 6.4 x 4.8 | | mm |
| $H_v$ | Optical sensitivity | | | 57'508 | | $\frac{LSB}{Lux/sec}$ |
| $A_{TOF}$ | TOF amplitude | dynamic range for distance measurement | 25 | | 1'200 | LSB |
| $A_{BW}$ | Grayscale amplitude | dynamic range for grayscale measurement | 0 | | 2'047 | LSB |
| $t_{INT}$ | Integration time | see Table 28 | 0.10 | | 52'600 | µs |
| $\Delta D_{DRIFT}$ | Distance drift | without DLL<br>with DLL and fully optimized system design | | tbd<br>tbd | | cm/K<br>cm/K |

Table 5: Optical characteristics

### 1.6. TOF and grayscale sensitivity

Typ. operational ratings, $T_A$ = +25ºC, modulation frequency 20MHz, integration time 103µs, unless otherwise stated

| Optical band | 640nm | 850nm | 940nm |
|---|---|---|---|
| Typ. TOF sensitivity $S_{TOF}$ | 203.5 $\frac{nW/cm^2}{LSB}$ | 155.0 $\frac{nW/cm^2}{LSB}$ | 203.5 $\frac{nW/cm^2}{LSB}$ |
| Typ. Grayscale sensitivity | | 61.5 $\frac{nW/cm^2}{LSB}$ | |

Table 6: TOF and grayscale sensitivity

### 1.7. Ambient-light suppression

Typ. operational ratings, $T_A$ = +25ºC, modulation frequency 20MHz, integration time 103µs, unless otherwise stated

| Optical band | 640nm, ± 27.5nm | 850nm, ± 32.5nm | 940nm, ± 30nm |
|---|---|---|---|
| min. suppression $E_{Sup}$ | 26.5 mW/cm² | 20.0 mW/cm² | 26.5 mW/cm² |
| Sunlight equivalent | 70 kLux | 69 kLux | 192 kLux |

Table 7: Ambient-light suppression

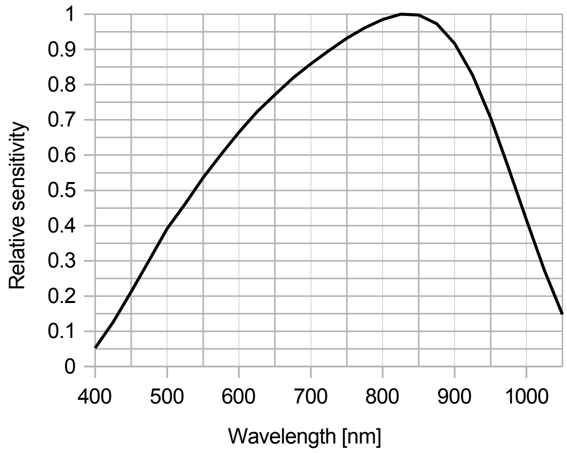



Figure 3: Relative spectral sensitivity ($S_\lambda$) sensitivity $S_{AC}$ vs. wavelength

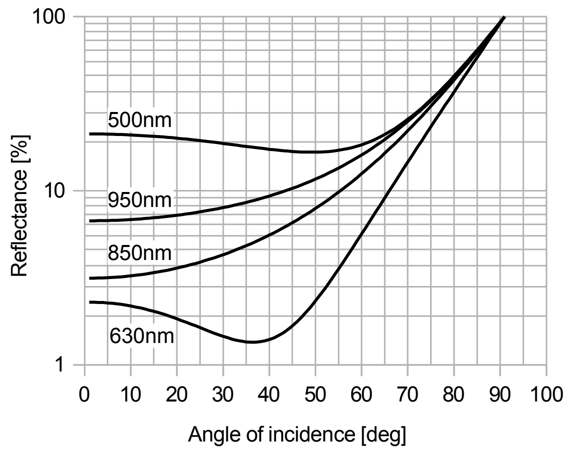



Figure 4: Reflectance vs. illumination angle (AOI)

## 2. Pin-out

### 2.1. Pin mapping

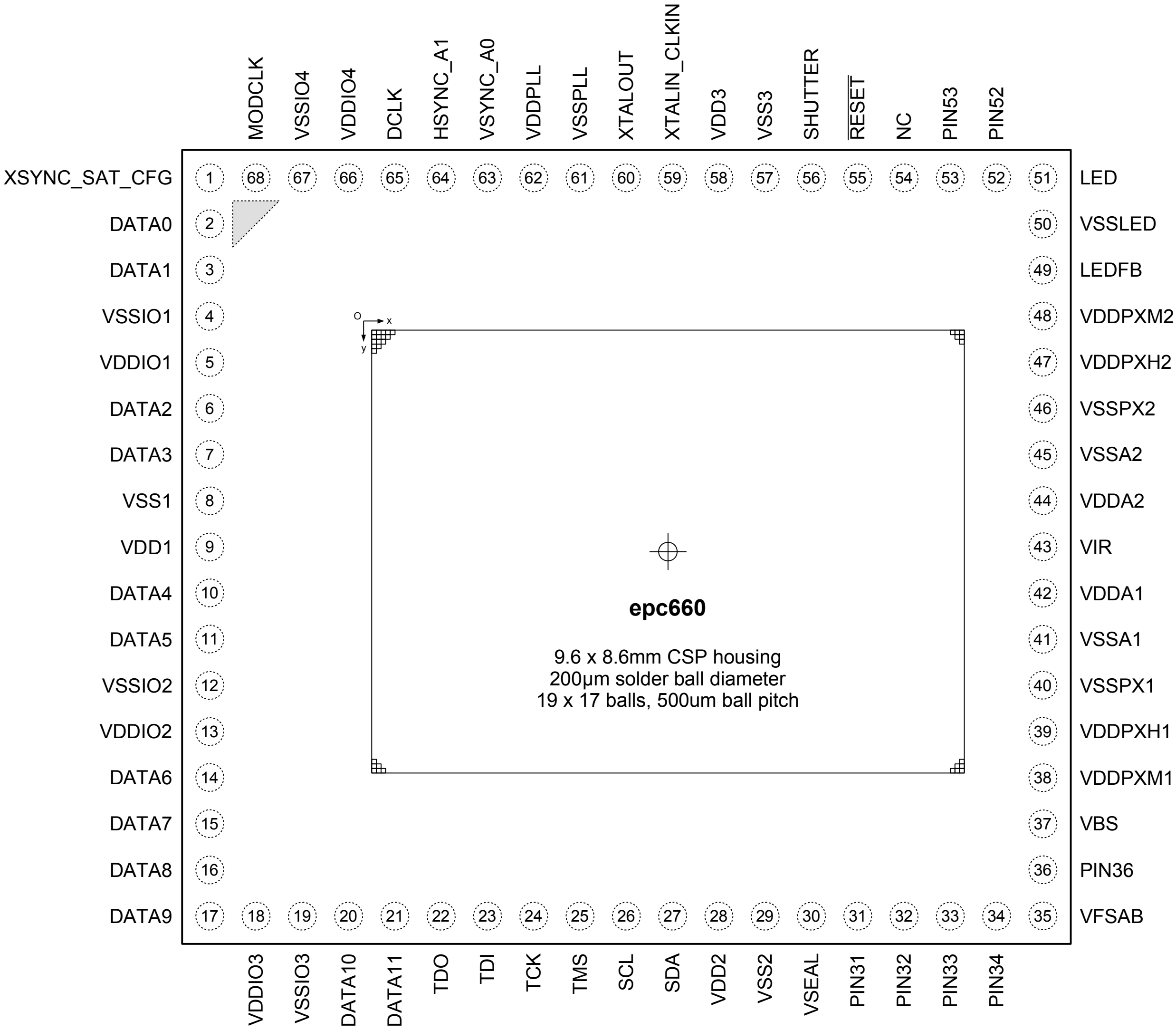



Figure 5: CSP pin mapping (top-view, solder balls are at the bottom, pixel field is at the top)

### 2.2. Pin list

| Pin No. | Pin name | Supply class $V_{SC}$ | Pin type | Rst func | Rst level | Description |
|---|---|---|---|---|---|---|
| **IO pins** | | | | | | |
| 2 | DATA0 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-0 (LSB) /<br>Strap 0: reserved |
| 3 | DATA1 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-1 /<br>Strap 1: reserved |
| 6 | DATA2 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-2 /<br>Strap 2: reserved |
| 7 | DATA3 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-3 /<br>Strap 3: reserved |
| 10 | DATA4 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-4 /<br>Strap 4: reserved |
| 11 | DATA5 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-5 /<br>Strap 5: reserved |
| 14 | DATA6 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-6 /<br>Strap 6: reserved |
| 15 | DATA7 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-7 /<br>Strap 7: reserved |
| 16 | DATA8 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-8 /<br>Strap 8: reserved |
| 17 | DATA9 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-9 /<br>Strap 9: reserved |
| 20 | DATA10 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-10 /<br>Strap 10: reserved |
| 21 | DATA11 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI high-speed output bit-11 (MSB) /<br>Strap 11: reserved |
| 65 | DCLK | $V_{DDIO}$ | DIO | IPD [1] | $V_{OL}$ | TCMI Data Clock output /<br>Strap 12: reserved |
| 63 | VSYNC_A0 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OH}$ | TCMI Vsync output /<br>Strap 13: $I^2C$ device address bit A0 |
| 64 | HSYNC_A1 | $V_{DDIO}$ | DIO | IPD [1] | $V_{OH}$ | TCMI Hsync output /<br>Strap 14: $I^2C$ device address bit A1 |
| 1 | XSYNC_SAT_CFG | $V_{DDIO}$ | DIO | IPD [1] | $V_{OH}$ | TCMI Xsync output /<br>TCMI Saturation flag output /<br>Strap 15: reserved |
| 26 | SCL | $V_{DDIO}$ | DIOD | I | $V_{IH}$ | $I^2C$ clock input [5] |
| 27 | SDA | $V_{DDIO}$ | DIOD | I | $V_{IH}$ | $I^2C$ data input/output [5] |
| 56 | SHUTTER | $V_{DDIO}$ | DI | PD | $V_{IL}$ | Shutter release input |
| 55 | $\overline{\text{RESET}}$ | $V_{DDIO}$ | DI | PD | $V_{IL}$ | Reset input (active low), 600kΩ int. pull-down [4] |
| 68 | MODCLK | $V_{DDIO}$ | DI | PD | | Modulator/demodulator external clock input.<br>If not used, connection to a test pad suggested. |
| 22 | TDO | $V_{DDIO}$ | DO | | $V_{OL}$ | JTAG test data output. Do not any electrical connection except to a test pad (suggested). |
| 23 | TDI | $V_{DDIO}$ | DI | PU | $V_{IH}$ | JTAG test data input. Do not any electrical connection except to a test pad (suggested). |
| 24 | TCK | $V_{DDIO}$ | DI | PD | $V_{IL}$ | JTAG test clock input. Do not any electrical connection except to a test pad (suggested). |
| 25 | TMS | $V_{DDIO}$ | DI | PU | $V_{IH}$ | JTAG test mode select input. Do not any electrical connection except to a test pad (suggested). |
| **Digital pins** | | | | | | |
| 59 | XTALIN_CLKIN | $V_{DDPLL}$ | AI | | | OSC input from external XTAL or Resonator /<br>CLKIN from external clock source |
| 60 | XTALOUT | $V_{DDPLL}$ | AO | | | OSC output (buffered) to ext. XTAL or Resonator |

Table 8: Pin list

| Pin No. | Pin name | Supply class $V_{SC}$ | Pin type | Rst func | Rst level | Description |
|---|---|---|---|---|---|---|
| **Analog pins** | | | | | | |
| 51 | LED | $V_{DDLED}$ | AOD | | $V_{LED}$ max | LED/LD driver open-drain output [3] |
| 49 | LEDFB | $V_{DDLED}$ | AI | | $V_{LEDFB}$ max | LED/LD driver feedback input [3] |
| 52 | PIN52 | --- | --- | | | Reserved. Do not any electrical connection except to a test pad (suggested). |
| 53 | PIN53 | --- | --- | | | Reserved. Do not any electrical connection except to a test pad (suggested). |
| 31 | PIN31 | $V_{DDPXH}$ | AI | | | Reserved. Do not any electrical connection except to a test pad (suggested). |
| 32 | PIN32 | $V_{DDPXH}$ | AI | | | Reserved. Do not any electrical connection except to a test pad (suggested). |
| 33 | PIN33 | --- | --- | | | Reserved. Do not any electrical connection except to a test pad (suggested). |
| 34 | PIN34 | --- | --- | | | Reserved. Do not any electrical connection except to a test pad (suggested). |
| 35 | VFSAB | $V_{DDPXH}$ | --- | | | Reserved. Connect to VSSPX. |
| 36 | PIN36 | $V_{DDPXH}$ | AI | | | Reserved. Connect to VSSPX. |
| **Supply pins, digital** | | | | | | |
| 5 | VDDIO1 | $V_{DDIO}$ | PWR | | | IO supply VDDIO |
| 13 | VDDIO2 | $V_{DDIO}$ | PWR | | | IO supply VDDIO |
| 18 | VDDIO3 | $V_{DDIO}$ | PWR | | | IO supply VDDIO |
| 66 | VDDIO4 | $V_{DDIO}$ | PWR | | | IO supply VDDIO |
| 9 | VDD1 | $V_{DD}$ | PWR | | | Digital supply VDD |
| 28 | VDD2 | $V_{DD}$ | PWR | | | Digital supply VDD |
| 58 | VDD3 | $V_{DD}$ | PWR | | | Digital supply VDD |
| 62 | VDDPLL | $V_{DDPLL}$ | PWR | | | PLL supply |
| 4 | VSSIO1 | $V_{DDIO}$ | GND | | | IO ground VSSIO |
| 12 | VSSIO2 | $V_{DDIO}$ | GND | | | IO ground VSSIO |
| 19 | VSSIO3 | $V_{DDIO}$ | GND | | | IO ground VSSIO |
| 67 | VSSIO4 | $V_{DDIO}$ | GND | | | IO ground VSSIO |
| 8 | VSS1 | $V_{DD}$ | GND | | | Digital ground VSS |
| 29 | VSS2 | $V_{DD}$ | GND | | | Digital ground VSS |
| 57 | VSS3 | $V_{DD}$ | GND | | | Digital ground VSS |
| 61 | VSSPLL | $V_{DDPLL}$ | GND | | | PLL ground |
| **Supply pins, analog** | | | | | | |
| 42 | VDDA1 | $V_{DDA}$ | PWR | | | Analog supply VDDA |
| 44 | VDDA2 | $V_{DDA}$ | PWR | | | Analog supply VDDA |
| 43 | VIR | $V_{IR}$ | PWR | | | Isolation supply |
| 37 | VBS | $V_{BS}$ | PWR | | | Bias supply |
| 39 | VDDPXH1 | $V_{DDPXH}$ | PWR | | | Pixel analog 2 supply VDDPXH |
| 47 | VDDPXH2 | $V_{DDPXH}$ | PWR | | | Pixel analog 2 supply VDDPXH |
| 38 | VDDPXM1 | $V_{DDPXM}$ | PWR | | | Pixel analog 1 supply VDDPXM |
| 48 | VDDPXM2 | $V_{DDPXM}$ | PWR | | | Pixel analog 1 supply VDDPXM |

Table 8 cont.: Pin list

| Pin No. | Pin name | Supply class $V_{SC}$ | Pin type | Rst func | Rst level | Description |
|---|---|---|---|---|---|---|
| 30 | VSEAL | $V_{SEAL}$ | PWR | | | Sealring supply |
| 41 | VSSA1 | $V_{DDA}$ | GND | | | Analog ground VSSA |
| 45 | VSSA2 | $V_{DDA}$ | GND | | | Analog ground VSSA |
| 40 | VSSPX1 | $V_{DDPX}$ | GND | | | Pixel analog ground VSSPX |
| 46 | VSSPX2 | $V_{DDPX}$ | GND | | | Pixel analog ground VSSPX |
| 50 | VSSLED | $V_{DDLED}$ | GND | | | LED/LD driver ground (return current) [2] |
| 54 | NC | --- | --- | | | Reserved. Do not any electrical connection except to a test pad (suggested). |

Table 8 cont.: Pin list

Notes:

1 DIO pin is configured as input with internal pull-down resistor enabled during $\overline{RESET}$ = 0 (active), only. As soon as $\overline{RESET}$ is driven to 1, pins stay in that state for another 4x osc_clk periods (see chapter 5.3.3., Strap pins), then the pin is configured as output, finally the internal pull-down resistor is disabled. From this moment on, the pin can be used for normal output function.

2 VSSLED is the dedicated, isolated GND pin for the LED/LD return-current from external circuitry. It must be connected to PCB GND plane together with the other VSSA GND pins.

3 LED output can be used to drive an external amplifier with an addition of a pull-up resistor. There are internal ESD protection diodes from LED output and LEDFB input to VDDA supply pins. Voltage at these pins must not exceed values in Table 1: Operating conditions and electrical characteristics.

4 RESET pin has a 600kΩ (typical) internal pull-down resistor. Therefore, this pin can be safely connected to a standard GPIO of a CPU which is initially high-Z or open-drain during power up sequence. Once the SW takes control, it can program this GPIO as output and drive 1 to release the RESET. The internal pull-down can be override by and external 10kΩ pull-up and a series capacitor to build a simple delayed power-on reset for evaluation/qualification purposes.

5 $I^2C$ pins SCL, SDA are according to $I^2C$ standards. They are $I^2C$ slave pins which need external pull-up resistors on the PCB. Values of R1 and R2 in the schematics are given only for indicative purposes and must be re-calculated according to the total capacitive load of all $I^2C$ slave/master devices and operating mode (FM or FM+) of the $I^2C$ (chapter 5.4.) in the application.

'Pin type' in Table 8 defines the following:

■ DI: Digital Input

■ DO: Digital Output

■ DIO: Digital Input/Output (bidirectional)

■ DIOD: Digital Input/Output (bidirectional), open-Drain

■ AI: Analog Input

■ AO: Analog Output

■ AOD: Analog Output, open-Drain

■ PWR: Supply

■ GND: Ground

'Rst. Func.' in Table 8 defines the function of IO pins during reset:

■ I: Input

■ PU: internal Pull-Up

■ PD: internal Pull-Down

■ IPD: Input with internal Pull-Down

'Rst. Level' in Table 8 defines the level of the IO pins during/after reset (chapter 5.3.)

### 2.3. Power domain separation and ESD protection

The epc660 chip has internally 10 different power domains and 6 ground references which are interconnected with ESD protection diodes. All pins are also equipped with ESD protection diodes. Figure 6 shows this functional circuit. The diodes have a breakthrough voltage of 0.3V. The designer has to take care that none of these diodes become conductive either at power-up, power-down or normal operation.

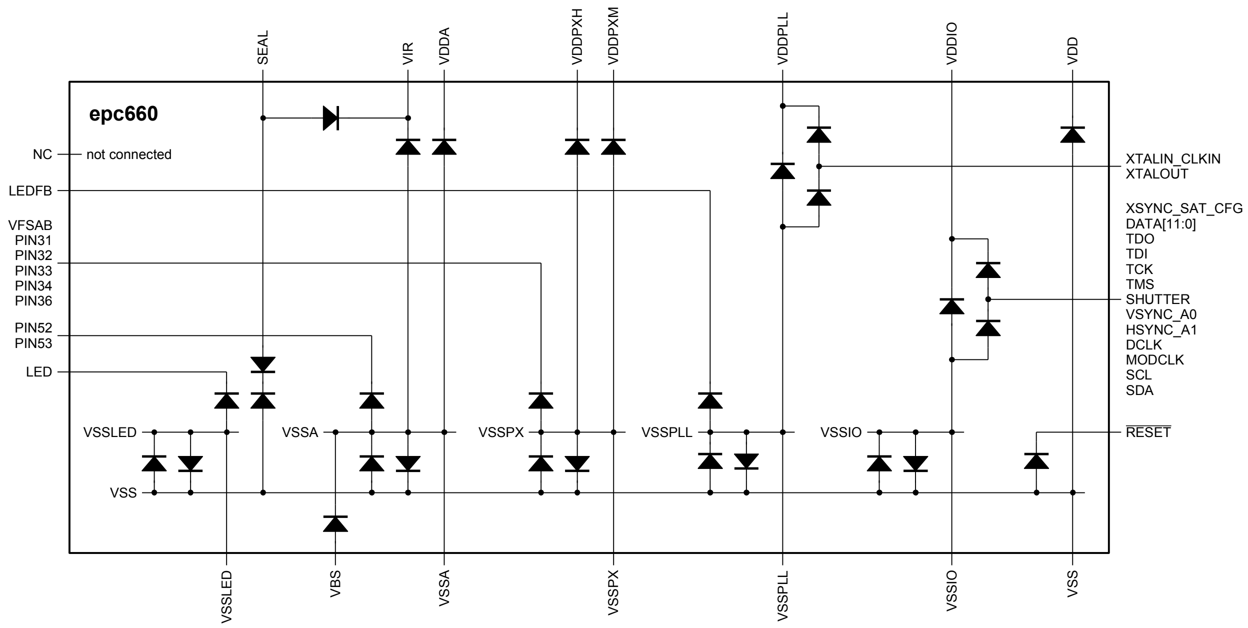



Figure 6: I/O pins and ESD protection diagram

Note:
Pin 51, NC can be used, to check if the chip is assembled the correct way on the PCB board. It is chip-internally not connected and will not show any ESD diodes.

# 3. Packaging and layout information

### 3.1. Mechanical dimensions

The center of the effective pixel field (320x240) is positioned with respect to the center of CSP pin 1 (XSYNC_SAT_CFG) with an offset of 4'900µm by 4'000µm (x,y). This point corresponds to intersection of middle of columns C163 - C164 and middle of rows R125 - R126 when mapped to pixel field coordinate system on the die (see Figure 29).

The packaging technology is chip scale packaging (CSP).

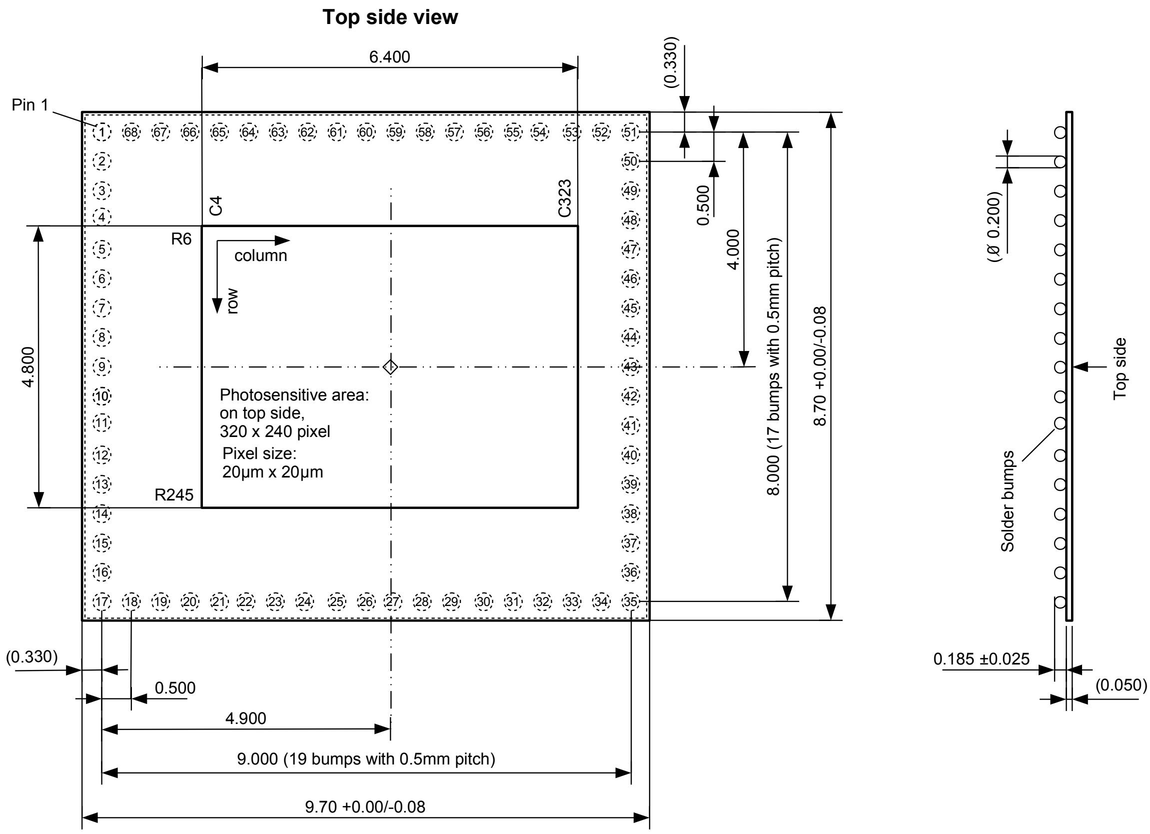
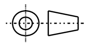


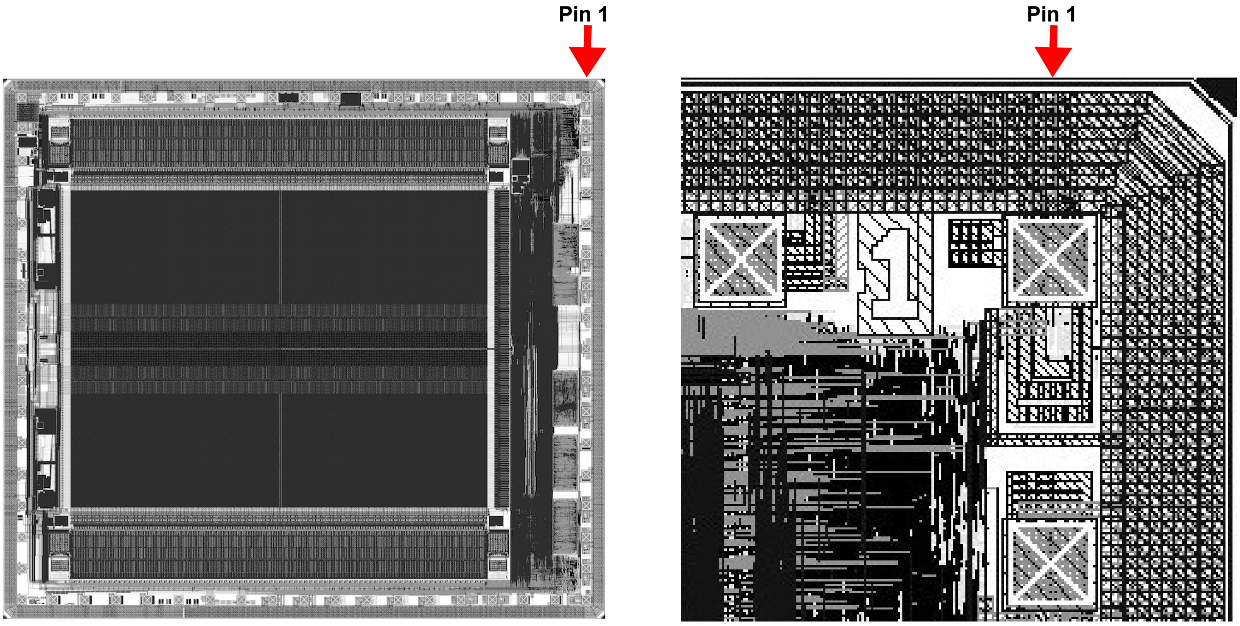

Figure 7: Mechanical dimensions

Notes:

- 
- all measures in mm
- not specified tolerances: ±0.001mm
- Dimensions in brackets: informal only
- Top side is illumination side

### 3.2. Pin1 marking

The following pictures shows the epc660 chip from the bottom side with view to the solder balls. Please note the location of pin 1.




epc660 chip from the solder ball side

Top right corner from the solder ball side

Figure 8: Pin 1 marking

### 3.3. Location of the photosensitive area

The photosensitive area is not marked neither on the front nor on the backside of the IC. As a visible reference, a metal ring of the IC can be used. From the solder ball side it is visible. Also from the front side (photosensitive area) it can be seen with a camera which is sensitive in the near infrared wavelength domain (950 .. 1'150nm).

### 3.4. PCB design and SMD manufacturing process considerations

Since the epc660 chip comes in a very small 68 pin chip scale package, the PCB layout should be made with special care. Since the silicon chip is small and light weight compared the the solder balls, it is highly recommended that all tracks to the chip should come straight from the side. A consequent symmetrical design is therefore highly recommended to achieve high production yield.

The pads and the tracks should also have exactly the same width and shall be covered by a solder resist mask in order to avoid drain of the solder tin alloy to the track.

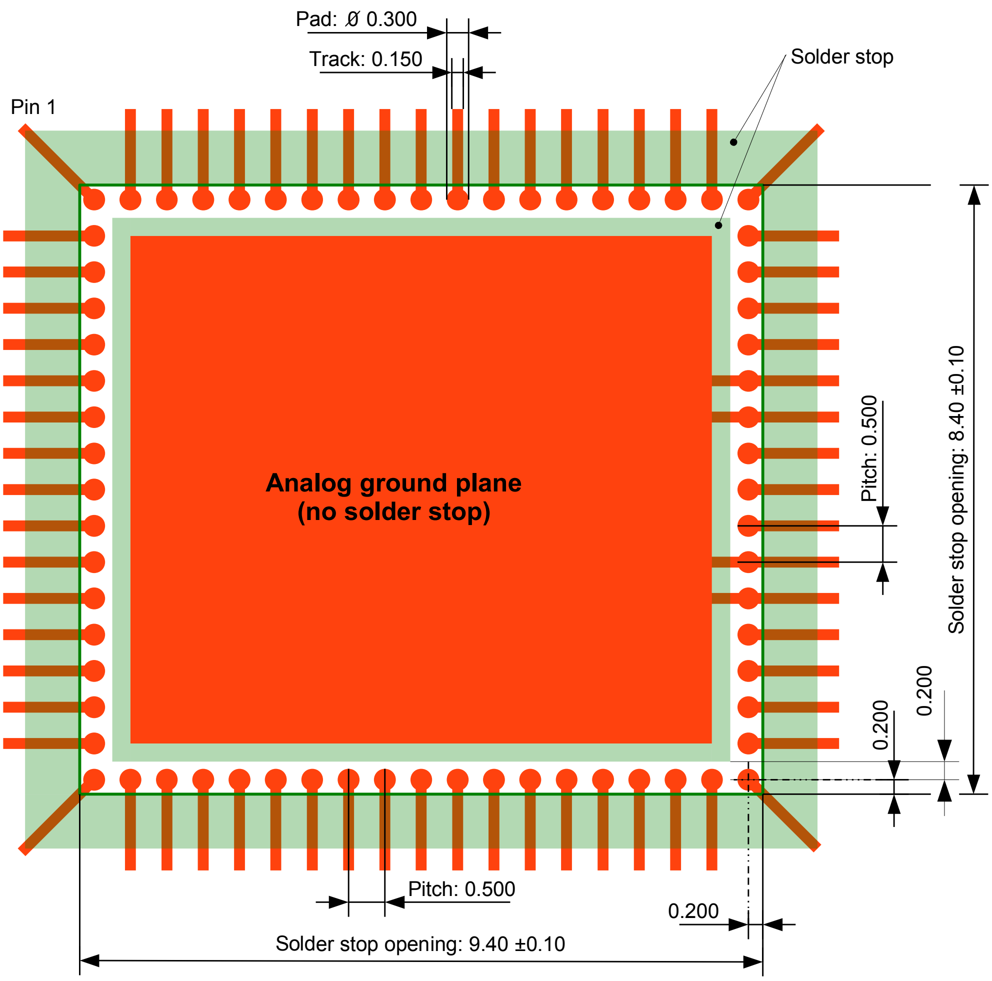



Figure 9: Recommended PCB layout (all measures in mm)

As shown in Figure 9, a ground plane shall be placed on the top PCB layer underneath the chip. This ground plane acts as a shield to suppress high frequency emission of fast interface signal lines. It is important that this plane is completely flat. Thus, the plane must not be scattered nor divided into sections. It should be rather full-faced and no via must be placed in this plane. Otherwise, the chip will pick up the surface and get bended.

Underfill of the components reduces stress to the solder pads caused by e.g. temperature cycling or mechanical bending.The thermal and mechanical fatigue will be reduced and the longterm reliability will be increased. Underfill and underfill selection is application specific. It shall follow JEDEC-STD JEP150: Stress-Test-Driven Qualification of and Failure Mechanisms Associated with Assembled Solid State Surface- Mount Components.

Please also refer to the application note AN04 - Assembly of Wafer Level Chip Scale (WL-CSP) Packages - which can be downloaded from the ESPROS website at www.espros.com/application-notes. Obeying the recommendations in the applications AN04, a high manufacturing yield can be achieved.

### 3.5. Soldering and IC handling

Since the chip is only 50µm thick, a careful handling during the surface mount assembly process shall be taken in order to avoid mechanical damage. In addition to that, careful PCB layout is needed in order to achieve reliable assembly results with a high yield. Please refer to the application note AN04 from epc which contains comprehensive information to these topics. This application note can be downloaded at www.espros.com/application-notes.

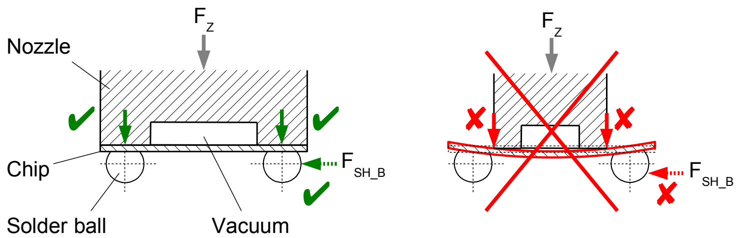
| Item | Description |
|---|---|
| Solder balls: Alloy | Sn96.5Ag3.0Cu0.5 (SAC305) |
| Soldering profile | For infrared or conventional soldering, the solder profile has to follow the recommendations of IPC/JEDEC J-STD020C (revision C and later) for lead-free assembly. |
| Soldering lead temperature | $T_L$ = 260 °C for 4s with a maximum gradient of 6 °C/s |
| Allowed cleaning agents | Isopropanol |
| Packaging technology | Bare-die CSP with underfill |
| Pick and place | <br><br>Figure 10: Pick & place; left: good nozzle, good picking; right: bad nozzle, bad picking<br><br>The nozzle needs a flat surface. It must not touch the pixel-field area.<br>It should not have any flashes where it touches the chip. The chip can get damaged.<br>The nozzle has to bring the $F_Z$ force evenly distributed in a vertical way to the center of the solder balls during picking and placing. Refer to Figure 10, left picture. It means that it should be large enough so that it picks the chip not just in the middle of the part but in the area above the solder balls.<br>Bending of the chip is not allowed. Refer to Figure 10, right picture<br>Keep attention having no shear forces to the solder balls during picking the chip from the tape pocket and/or placing.<br>Maximum vertical force $F_Z$ per chip: $\leq$ 0.8 N.<br>Maximum shear force $F_{SH\_B}$ per ball: $\leq$ 0.5 N.<br>Preferred is forceless picking with vacuum only.<br>It is also highly recommended to use a „kiss-and-goodbye“ concept to place the component. It means that the nozzle should blow the part away when it does the placing. |

Table 9: IC handling specification

# 4. Ordering Information

| Part Number | Part Name | Package | RoHS compliance |
|---|---|---|---|
| P100 183 | epc660-CSP68 | CSP68 | Yes |
| P100 244 | epc660 Card Edge Connector Carrier | PCB 37.25 x 36.00 mm | Yes |

Table 10: Ordering Information

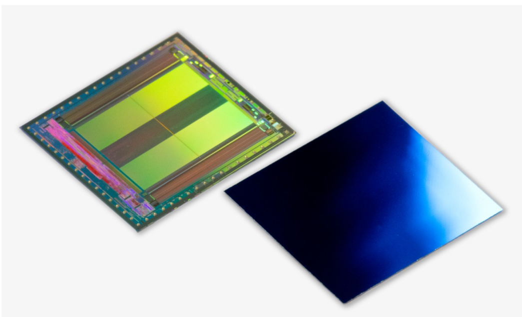


Figure 11: epc660-CSP68
(bottom and top side)

Figure 12: epc660 Card Edge Connector Carrier

# 5. Functional Description

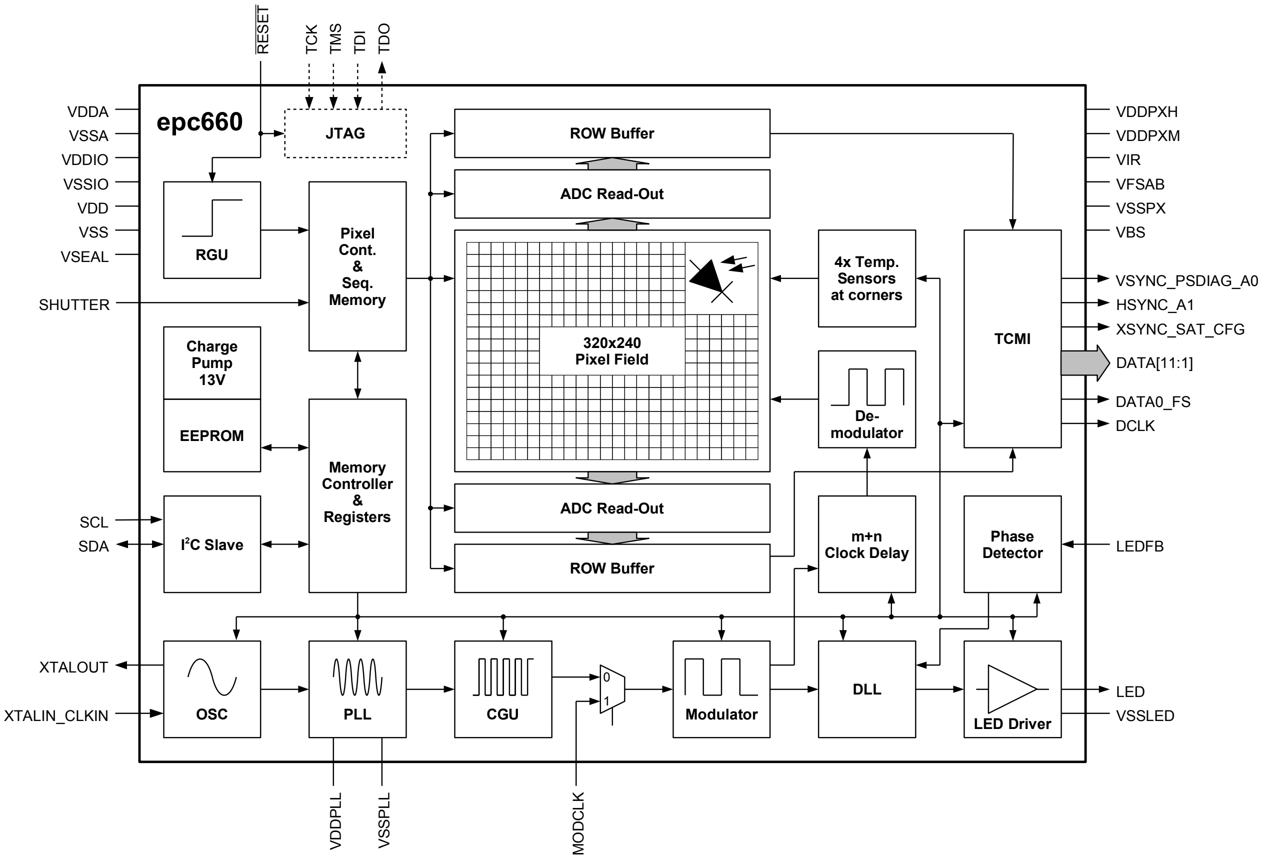



Figure 13: Block diagram

Figure 13 shows the relations between the functional blocks with signal flow diagram.

Based on the clock and mode setup of the chip, the modulation signal is generated and output over the LED driver to the external IR LED. Current flowing through the LED output pin can be monitored via LEDFB pin. This information is used for compensating delays generated due to temperature changes and aging of the LED in the system. Instead of an LED, also Laser diodes (LD) can be used.

The compensation is used for delaying the demodulation signal going into the pixel field.

In the pixel field, the reflecting IR light returning from the object is captured and converted to electrons which are transferred into two storage gates within each pixel (MGA and MGB) depending on the phase information of the demodulation signal.

The parallel ADC blocks on top and bottom pixel field efficiently convert and transfer the phase information to the digital domain where it will be formatted and finally output via the TCMI for external distance calculation.

### 5.1. Typical Application Diagram

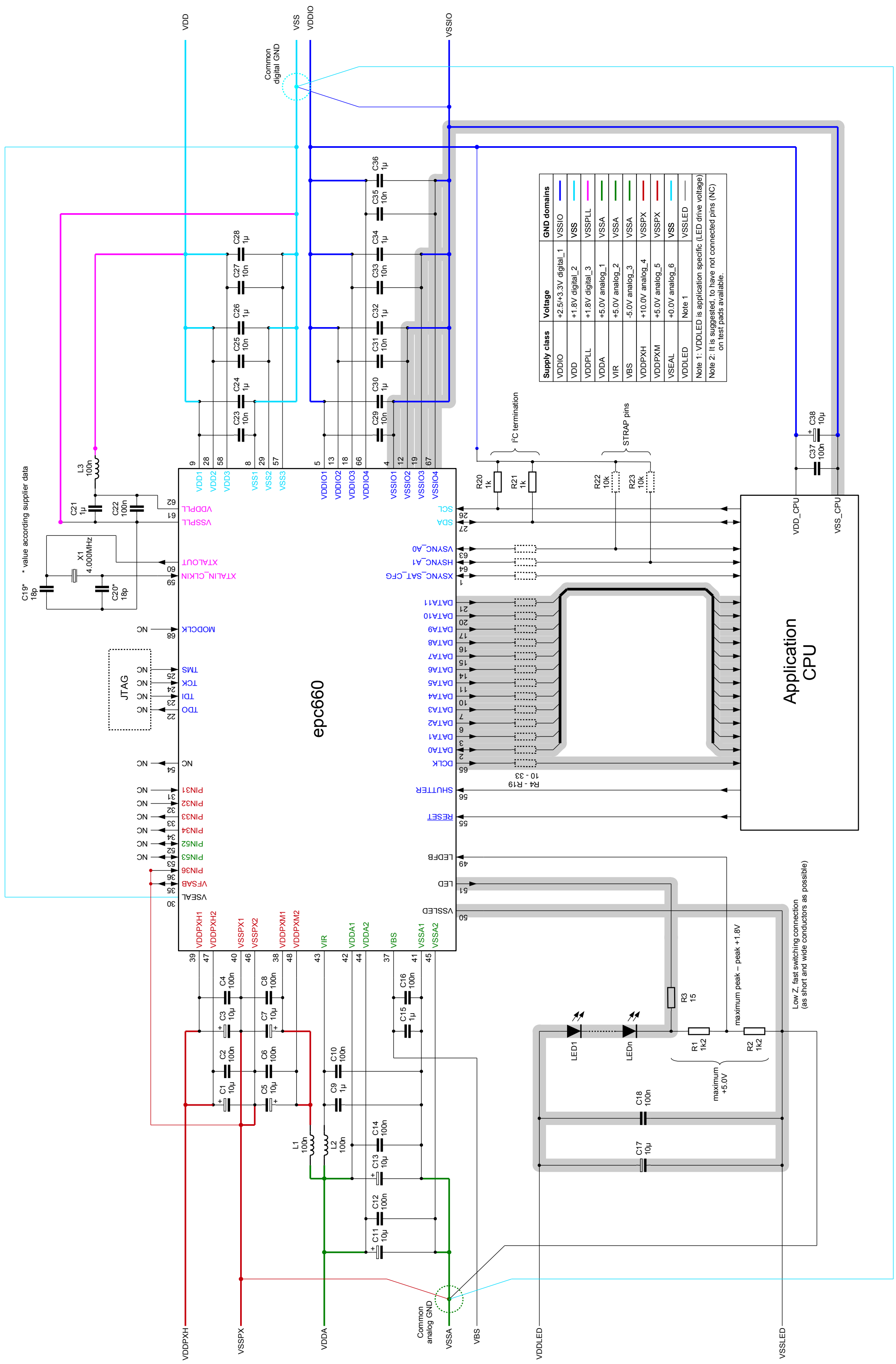



Figure 14: Typical Application Diagram

epc660 is supplied with four positive (+1.8V, +2.5/3.3V, +5V, +10V) and one negative (-10V) external DC supply voltage levels (Figure 14).

- Decoupling capacitors must be placed next to each supply pin pair in order to minimise the instantaneous voltage drop on internal supply rails due to fast switching high-speed signals (Table 11). They help reducing switching noise on internal/external supply rails. Therefore, they minimize the impact on system's optical/electrical performance.

+1.8V is used for supplying the digital logic (VDD), the on-chip oscillator OSC and the phase-look-loop PLL (VDDPLL).

- The digital logic creates some internal switching noise on VDD.
- When same supply is shared together with OSC and PLL, their supply wiring must be separated from the digital wires and physically isolated from each other. These supplies are marked in the application diagram as VDD and VDDPLL, respectively (Figure 14). A good practice is inserting on the PCB a series inductance of 100nH between them close to the supply source, then creating separate supply islands for both on the board. The XTAL/OSC and PLL are critical parts of the chip which directly impacts the optical system performance (i.e. distance calculation). Therefore, if the +1.8V supply is too noisy, it is recommended for the application to have a dedicated silent +1.8V supply only for OSC and PLL.

+2.5/3.3V is used for supplying the high-speed IO pins and the slow I$^2$C pins (VDDIO).

- High speed IO pins toggle at 20/40/80MHz during data transfer, hence generating continuously switching noise (much more dominant than the digital noise). Therefore VDDIO supply wires and layers must be carefully designed and isolated in a separate supply island on the PCB.
- I$^2$C pins external pull-up resistors can share the same supply on the PCB.
- As the name implies, the IO supply voltage can be one of the two +2.5V, +3.3V or in between. The specific IO voltage depends on the choice of the application CPU's IO voltage and other system supply requirements. It is not recommend to change this voltage on the fly when the TCMI or I$^2$C interfaces are running. When the application needs power saving during system idle periods, it can be scaled from +3.3V down to +2.5V only after frame acquisition is stopped and both interfaces are completely inactivated. It can be increased back to +3.3V before re-activating the chip for frame acquisition, accessing I$^2$C or TCMI interface. Note that, voltage scaling must be done in a controlled way having both application CPU's and epc660's IO voltages at the same time at the same level.
- Some CPU's can also work at +2.5V IO voltage but with reduced IO toggle speed. In such situations, application must make sure that the right TCMI DCLK frequency is set, both on the epc660 and on the CPU itself.

+5V is used for supplying three different analog blocks of the chip, pixel field drivers (VDDPM), ADC readout circuitry (VDDA) and isolation (VIR). Refer to Figure 14).

- Internal supplies of these three blocks are decoupled from each other. Therefore, their supply pins must also be decoupled on the PCB. It is recommended to insert a series inductance of 100nH between them close to the supply source on the PCB. A separate supply island on the PCB is also recommended.

VBS is used for biasing the the pixel field.

- The use of a stable supply source with a low ripple is recommended. There is no switching or active internal circuit working on that. Therefore this input can be simply decoupled as indicated application diagram (Figure 14), and does not need a dedicated supply island on the PCB.

A 4MHz quartz crystal or a ceramic resonator together with two 18pF load capacitors must be connected to XTALIN_CLKIN and XTALOUT pins in order to use internal oscillator OSC as time base for the epc660. The capacitor value has to be selected according the crystal or resonator supplier's recommendation. The frequency accuracy and stability are directly related to the distance readings. Alternatively an external clock source can be connected to XTALIN_CLKIN pin, leaving XTALOUT unconnected (chapter 5.2.).

The optional MODCLK input can be used for user controlled/modulated clock for both the LED driver and the pixel field demodulator.

SCL, SDA are I$^2$C slave pins which need external pull-up resistors on the PCB (see also VDDIO supply). Values of R20 and R21 are given only for indicative purposes and must be re-calculated according to the total capacitive load of all I$^2$C slave/master devices and the operating mode FM or FM+ of the I$^2$C (chapter 5.4.) in the application.

VSYNC_A0, HSYNC_A1, XSYNC_SAT_CFG, DATA[11:0], DCLK high-speed TCMI signals (chapter 5.8.), SHUTTER and $\overline{\text{RESET}}$ control signals toggle in the VDDIO range.

- To minimize the skew, the high-speed *SYNC, DATA[11:0], DCLK signals wires must be routed equal in impedance and length less than 10cm long with less than 10mm difference on the PCB. As they are toggling all the time, they can be separated with ground wires on the side adjacent to other signals/supply lines, routed with enough distance from other sensitive signal wires on the board. Series termination resistors R4 … R19 (10 … 33Ω) are needed at high-speed outputs to control the slew.
- They must be directly connected to the application CPU. In some cases $\overline{\text{RESET}}$ can also be driven from power management device.
- There are optional pull-up resistors R22 and R23 (10kΩ) to set initial values of some configuration registers during start up of the chip. Such outputs pins are called strap pins. They are scanned one time immediately after $\overline{\text{RESET}}$ is released (chapter 5.3.3.). Two LSB of the I$^2$C device address bits are programmable via these strap pins during start-up (chapter 5.3.). When there is only one slave and one master on the I$^2$C bus as in the application diagram (Figure 14), these resistors can be omitted.

The LED pin is the internal open-drain IR LED/LD driver output. When the driver is active (on), the IR LED/LD on-current flows through the power resistor R3 into the LED pin, through the driver and comes out of the chip on the VSSLED ground pin.

- The LED pin toggles up to 20MHz or according to the MODCLK clock with a current maximum of 350mA limited by the resistor R3.
- The number of IR LEDs depends on the level of the LED supply voltage and the turned-on forward voltage drop of the IR LEDs.
- This signal creates a lot of ground noise. Therefore, VSSLED pin is decoupled from the other analog grounds internally. It must be shorted with the other analog ground pins with a low-ohmic connection as short as possible on the PCB. In this way, there will be minimal voltage differences in the ground planes of the board.
- The LED supply line must be isolated properly from any analog supply on the PCB to minimize noise coupling from the LED drivers.

The LEDFB pin is the amplifier input for the LED feedback signal. The easiest way is to pick it from a voltage divider R1 and R2 following the IR LEDs on the PCB.

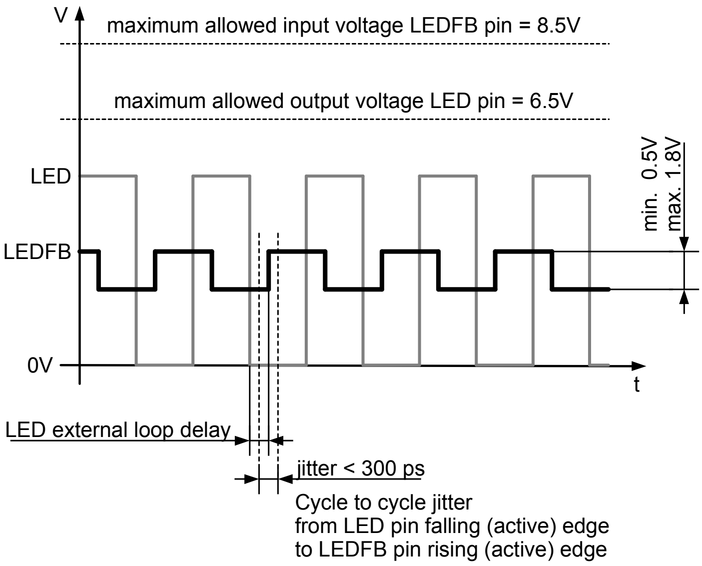



Figure 15: LED and LEDFB pin signal restrictions

TCK, TMS, TDI, TDO are standard JTAG controller pins (Additional information upon request)

It is recommended having “not connected pins” (NC) on test pads available. It helps e.g. to check after assembly for correct orientation of the chip or for short-cuts.

To guarantee the proper function of the chip, the remaining, not here listed pins have to be connected according Figure 14.

| Part No. | Description | Pin No. | Value | | | Toler-ance | Supply class $V_{SC}$ | Comments |
|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | | | |
| C1, C3 | VDDPXH | 46 – 47, 39 - 40 | 10 µF | | | ±20% | $V_{DDPXH}$ | Low ESR |
| C5, C7 | VDDPXM | 46 – 48, 38 - 40 | 10 µF | | | ±20% | $V_{DDPXM}$ | Low ESR |
| C11, C13 | VDDA | 44 – 45, 41 - 42 | 10 µF | | | ±20% | $V_{DDA}$ | Low ESR |
| C9 | VIR | 41 - 43 | 1 µF | | | ±20% | $V_{IR}$ | Ceramic X7R |
| C15 | VBS | 37 - 41 | 1 µF | | | ±20% | $V_{BS}$ | Ceramic X7R |
| C21 | VDDPLL | 61 - 62 | 1 µF | | | ±20% | $V_{DDPLL}$ | Ceramic X7R |
| C24, C26, C28 | VDD | 8 – 9, 28 – 29, 57 - 58 | 1 µF | | | ±20% | $V_{DD}$ | Ceramic X7R |
| C30, C32, C34, C36 | VDDIO | 4 – 5, 12 – 13, 18 – 19, 66 - 67 | 1 µF | | | ±20% | $V_{DDIO}$ | Ceramic X7R |
| C2, C4 | VDDPXH | 46 – 47, 39 - 40 | | 100 nF | | ±20% | $V_{DDPXH}$ | Ceramic X7R |
| C6, C8 | VDDPXM | 46 – 48, 38 - 40 | | 100nF | | ±20% | $V_{DDPXM}$ | Ceramic X7R |
| C10 | VIR | 41 – 43 | | 100 nF | | ±20% | $V_{IR}$ | Ceramic X7R |
| C12, C14 | VDDA | 44 – 45, 41 - 42 | | 100 nF | | ±20% | $V_{DDA}$ | Ceramic X7R |
| C16 | VBS | 37 - 41 | | 100 nF | | ±20% | $V_{BS}$ | Ceramic X7R |
| C22 | VDDPLL | 61 - 62 | | 100 nF | | ±20% | $V_{DDPLL}$ | Ceramic X7R |
| C23, C25, C27 | VDD | 8 – 9, 28 – 29, 57 - 58 | | 10 nF | | ±20% | $V_{DD}$ | Ceramic X7R |
| C29, C31, C33, C35 | VDDIO | 4 – 5, 12 – 13, 18 – 19, 66 - 67 | | 10 nF | | ±20% | $V_{DDIO}$ | Ceramic X7R |
| C19, C20 | XTAL | 59, 60 | --- | 18 pF [2] | --- | ±20% | $V_{DDPLL}$ | Ceramic NPO |
| L1 | VDDPXM | --- | | 100 nH | | ±20% | $V_{DDPXM}$ | Inductor |
| L2 | VIR | --- | | 100 nH | | ±20% | $V_{IR}$ | Inductor |
| L3 | VDDPLL | --- | | 100 nH | | ±20% | $V_{DDPLL}$ | Inductor |
| X1 | XTAL | 59 - 60 | --- | 4 MHz | --- | ±100ppm | $V_{DDPLL}$ | Quartz / Resonator |
| R4 - R19 | Bus termination | | 0 Ohm | 10 Ohm | 33 Ohm | ±20% | $V_{DDIO}$ | Resistors |
| R20, R21 | I²C pull-up | | | 1 kOhm | | ±20% | $V_{DDIO}$ | Resistors |
| R22, R23 | I²C address | | | 10 kOhm | | ±20% | $V_{DDIO}$ | Resistors |

Table 11: Values of component related to epc660 chip (Figure 14)

Notes:

1 All other components are application specific.

2 The capacitor value has to be selected according the crystal or resonator supplier's recommendation.

### 5.2. External Clock

epc660 can be driven by an external 4MHz clock source connected to XTALIN_CLKIN input pin. XTALOUT output pin left unconnected. Input clock signal levels must match $V_{DDPLL}$/$V_{SSPLL}$ supply levels (Table 1).

When the external clock source is coming from +2.5/3.3V voltage domain, a simple resistor divider circuit can be deployed to adjust the voltage levels. The following resistor values R26/R27 are recommended: 1.2kΩ/2.7kΩ with +2.5V voltage domain; 1kΩ/1kΩ with +3.3V voltage domain (see Figure 16).

XTALOUT is not designed to drive external loads. Therefore, it is not recommended to use XTALOUT as a “buffered clock source” for another device in the application circuit.

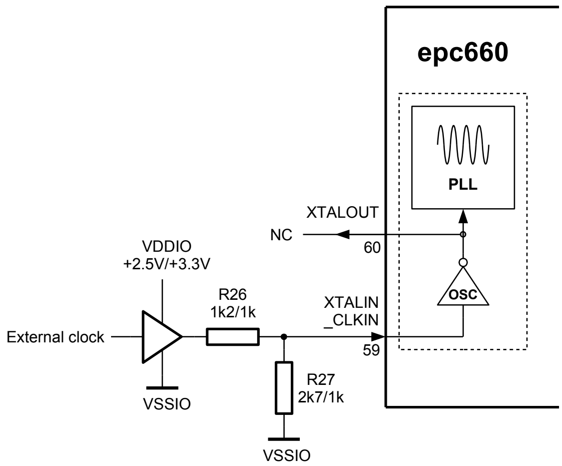



Figure 16: Resistor divider to adjust external clock voltage levels to XTALIN_CLKIN

**IMPORTANT:** The optical performance of the chip directly depends on the input clock precision/stability; therefore, only a clean, stable clock must be used as external clock source.

### 5.3. Supply, Reset and Start-up Options

#### 5.3.1. Supply voltages and external reset

During the power-up sequence, $V_{DD}$ and $V_{DDPLL}$ supplies (Figure 17) must be applied at the same time to the epc660. $V_{DDIO}$ can be applied either at the same time or after $V_{DD}$ and $V_{DDPLL}$ supplies become stable. In a system where $V_{DDIO}$ voltage is connected in parallel to application CPU IO supply pins (see Figure 14), $V_{DD}$ and $V_{DDPLL}$ can be generated by a linear regulator directly from $V_{DDIO}$ supply. In this case, all these three supplies ramp together.

$V_{DDA}$, $V_{IR}$, $V_{DDPXM}$ and $V_{DDPXH}$ supplies must be applied as a second group, after all $V_{DD}$, $V_{DDPLL}$ and $V_{DDIO}$ supplies become stable.

The negative supply $V_{BS}$ must be applied after all positive supplies reached their rated levels.

The $\overline{\text{RESET}}$ input must be kept active (low) while all positive voltages are ramping-up in order to guarantee proper reset of all internal circuits. As soon as rated positive levels are reached, $\overline{\text{RESET}}$ can be released (high). On the other hand, there is no strict relation between $V_{BS}$ level and $\overline{\text{RESET}}$ timing. Application must guarantee that the $V_{BS}$ supply is turned on and reached its negative rated level before the chip is triggered via the SHUTTER for frame acquisition. In case of an external clock is applied at XTALIN_CLKIN instead of a crystal/resonator is used with on-chip OSC, clock must be present before $\overline{\text{RESET}}$ is released.

**IMPORTANT:**

- It is possible to shutdown entire supplies for a very low standby current. In that case, first $\overline{\text{RESET}}$ must be driven low, then supplies must be turned off in the reverse order. Refer for details to chapter 7.7., Power saving options.
- $V_{DDA}$, $V_{IR}$, $V_{DDPXM}$ and $V_{DDPXH}$ supplies must be never kept on while turning off $V_{DD}$, $V_{DDPLL}$ and $V_{DDIO}$. Such condition may create a permanent damage to the chip.

For rated values of supply voltages, refer to Table 1: Operating conditions and electrical characteristics.

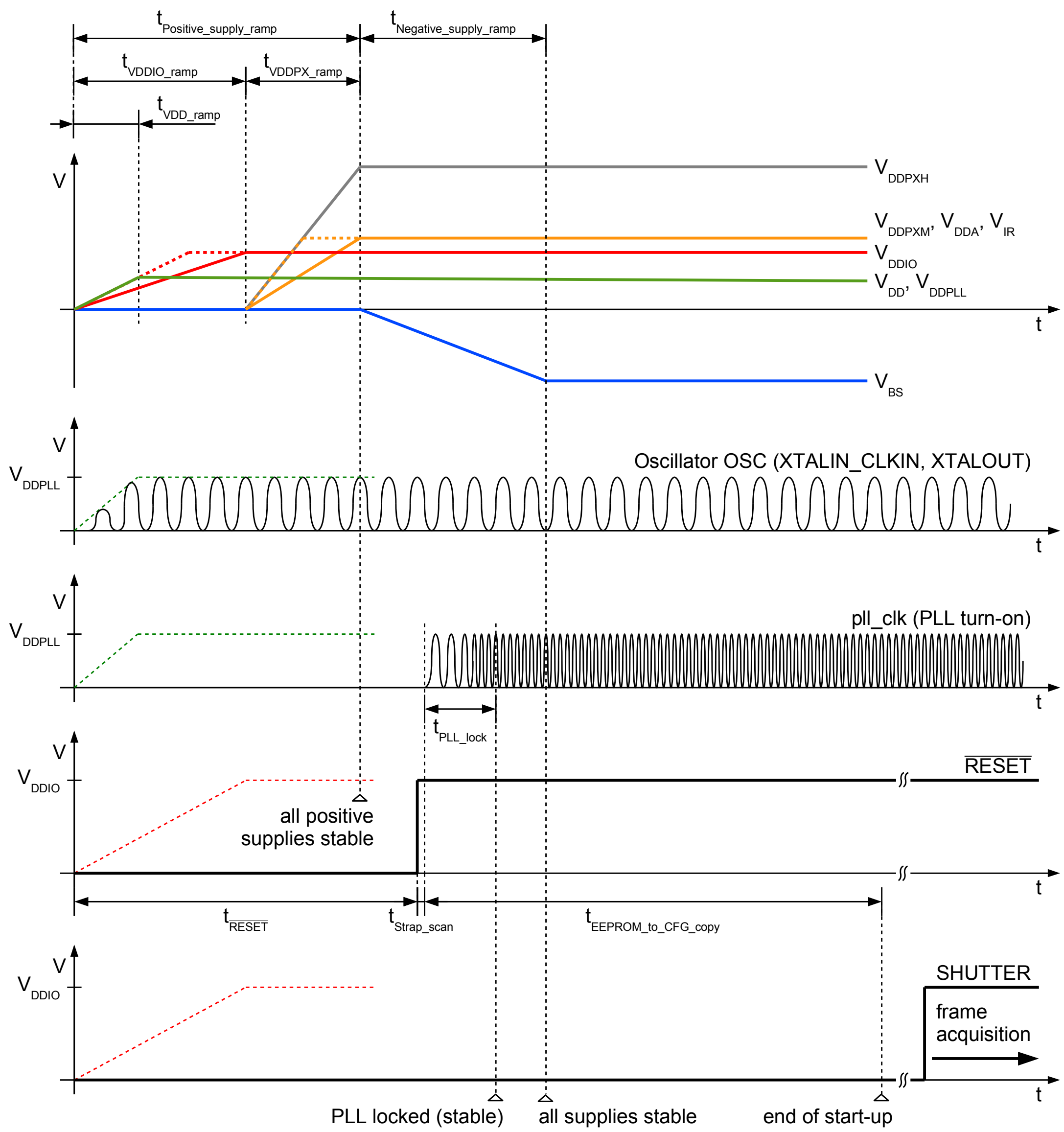



Figure 17: Start-up and reset sequence during power-up (normal mode)

#### 5.3.2. Software reset via I²C

The epc660 can be software reset by sending I²C general call reset command (chapter 5.4.4.). Reset behaviour is similar to RESET pin. The chip is resetting including all analog and digital blocks at the same time.

#### 5.3.3. Strap pins

The epc660 has output pins with dual/alternative functionality for PCB level flexible start-up configuration changing, called 'strap pins'. $\overline{\text{RESET}}$ release is followed by a very short strap pin scanning step. The chip programs all of its strap pins as inputs with internal weak pull-downs enabled for 4 osc_clk periods. If there is no external pull-up resistor connected, the corresponding strap pin will be scanned as logic 0 due to internal pull-down resistor. If there is an external pull-up resistor connected (Figure 14), it will override the internal pull-down and corresponding pin will be scanned as logic 1. After the strap scan period, pins are programmed back as outputs so that they can be used for their main function. Strap pins and their definitions are listed below (Table 12).

| Pin | Strap pin | Definition |
|---|---|---|
| XSYNC_SAT_CFG | 15 | reserved |
| HSYNC_A1 | 14 | Set A1 bit of 7-bit I²C slave device address (section 5.4.1.). |
| VSYNC_A0 | 13 | Set A0 bit of 7-bit I²C slave device address (section 5.4.1.). |
| DCLK, DATA11 … DATA0 | 12, 11 … 0 | reserved |

Table 12: Strap pin definition

**IMPORTANT:**
For having the strap pin function correctly, there should be only one 10kΩ pull-up resistor per pin be active on the line during strap-scan phase. No other pull-down/pull-up resistor, neither on the board nor in the pins of the application CPU shall be added or enabled, respectively. Destination pins in the receiving device (application CPU) must be configured always as inputs during the start-up and never programmed as outputs at any time. If the receiving device by default enables its internal pull-down/up resistors during start-up phase, the application SW must disable them before the reset is released and until the end of the strap-scan phase.

#### 5.3.4. Start-up (Clock, PLL turn-on and EEPROM copy)

The epc660 starts using either the internal 4MHz oscillator OSC with a crystal/resonator (Figure 14) or an external 4MHz clock, followed by an EEPROM copy sequence in parallel to the PLL turn-on phase. This is the factory default configuration.

Several configuration registers are modified by copying the EEPROM content (Figure 74 and Table 42, i.e. overwrite reset values). The EEPROM copy step takes 340µs after the $\overline{\text{RESET}}$ is released. Note, the PLL lock time (stable < 32µs) is relatively faster compared to the slower EEPROM copy.

### 5.4. I$^2$C Slave

The I$^2$C-bus interface allows accessing the RW registers and the programming of the EEPROM registers which stores the configuration parameters. It is the only interface through which the configuration registers can be accessed (Figure 74, Table 41 and Table 42). It works as a slave device according to the I$^2$C specification (refer to chapter 9.2.) with a transfer rate of up to 400 kbit/s in Fast Mode (FM) or 1M-bit/s in Fast Mode plus (FM+). The I$^2$C master such as an external CPU can set the transfer speed simply by driving the SCL input at that frequency (up to 1MHz), therefore there is no prior register configuration or setting necessary.

I$^2$C specification is supported in epc660 with following remarks/exceptions:

- Only 7-bit addressing is supported.
- Clock stretching is supported.
- General call address: By transmitting 0x00 followed by 0x06 (issues software reset) or transmitting 0x00 followed by 0x04 (device address reload), the programmable part (A0, A1) of the I$^2$C address pins is overwritten by the initially scanned value through strap pins during start-up or reset phase.
- Software reset is supported.
- Other uses of I$^2$C bus are not supported.

#### 5.4.1. Device addressing

The epc660 7-bit I$^2$C device address is hard-wired to the value shown below in Figure 18. Two address bits A0, A1 can be optionally initialized as 1 through strap pins (chapter 5.3.3.). In a typical single-camera 3D TOF imager application in which epc660 is directly connected as a single I$^2$C slave to a single I$^2$C master, the strap pins can be used without any external pull-up resistors. In this case, the device address is set after reset default as 0100000. In a multi-camera application with up to 4 epc660 devices connected on the same I$^2$C bus as slaves or together with other I$^2$C slaves talking to a single I$^2$C master, external pull-up resistors can be utilized on the strap pins to initialize different I$^2$C device addresses in order to correctly identify different epc660 slaves on the bus.

MSB LSB

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | A1 | A0 | **R/$\overline{W}$** |

Figure 18: Device address through I$^2$C

#### 5.4.2. I$^2$C bus protocol notation

The following notation is used:

- S **S**TART condition
- P STO**P** condition
- A **A**cknowledge last byte (ACK)
- $\overline{A}$ **N**ot-Acknowledge last byte (NACK)
- Shaded part of protocol: transmitted by master
- Unshaded part of protocol: transmitted by slave

### 5.4.3. I²C bus timing

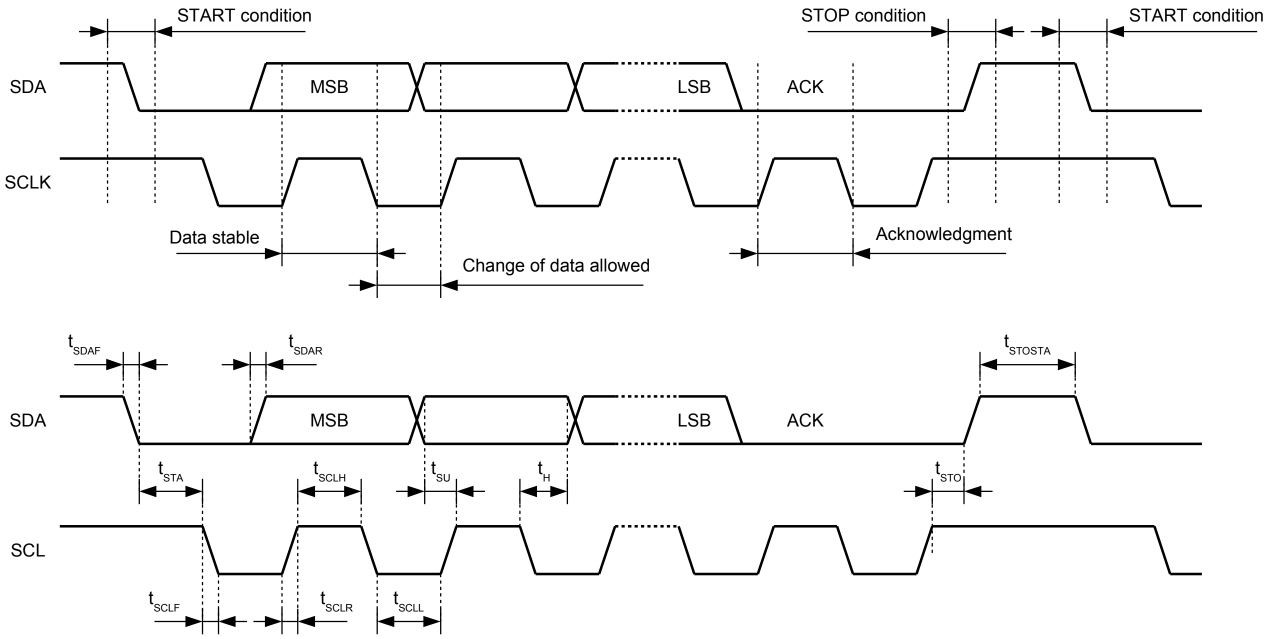



Figure 19: I²C bus timing
top : Basic communication sequence
below: Timing parameters

| Symbol | Parameter | Min. | Max. | Units |
|---|---|---|---|---|
| $f_{SCL}$ | I²C data rate | | 1 | Mbit/s |
| $t_{SCLL}$ | SCL clock low time | 0.5 | | µs |
| $t_{SCLH}$ | SCL clock high time | 0.26 | | µs |
| $t_{SU}$ | SDA setup time | 50 | | ns |
| $t_{H}$ | SDA hold time | | 0 | ns |
| $t_{SDAR}$<br>$t_{SCLR}$ | SDA and SCL rise time | | 120 | ns |
| $t_{SDAF}$<br>$t_{SCLF}$ | SDA and SCL fall time | | 120 | ns |
| $t_{STA}$ | Start condition hold time | 0.26 | | µs |
| $t_{STO}$ | Stop condition setup time | 0.26 | | µs |
| $t_{STOSTA}$ | Stop to start condition time (bus free) | 0.5 | | µs |
| $C_{b}$ | Capacitive load for each bus line | | 550 | pF |
| $t_{SP}$ | Pulse width of the spikes that are suppressed by the analog filter | | 50 | ns |

Table 13: I²C bus timing: Timing parameters (FM+)

### 5.4.4. General calls

epc660 supports two general call commands:

- (0x00, 0x06) issues a software reset, same behavior like with $\overline{\text{RESET}}$ pin.

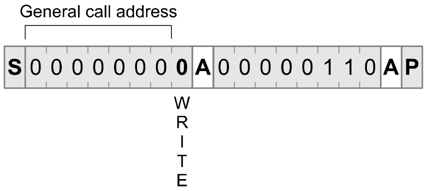



Figure 20: Software reset through I²C

- (0x00, 0x04) activates the I²C address stored in register 0xCA. Note that the the values of A0 and A1 cannot be changed by software. Therefore, this general call command only works for bits 2 to 6 of register 0xCA (chapter 5.3.3).

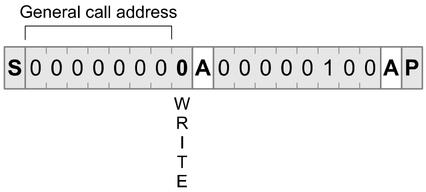



Figure 21: Device address A1, A0 reload through I²C

### 5.4.5. Write access

The epc660 I²C interface offers single-byte write access and multi-byte write access, where the former is basically a subset of the latter.

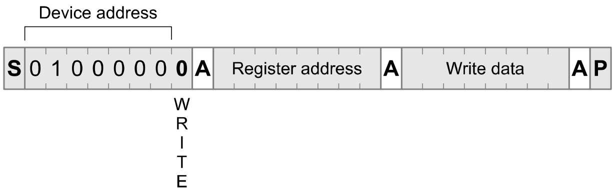



Figure 22: Single-byte Write access through I²C

During a single-byte write, only one register is written. After the device address is transmitted, the master has to transmit the register address and the write data in two I²C data packets (Figure 22). The access is terminated by a STOP condition.

During a multi-byte write operation, the master transmits the device address and the address of the first register to be written. All subsequent bytes until the STOP condition are interpreted as write data packets by the epc660 (Figure 23). The write address pointer is incremented internally. It is illegal to transmit so many data packets that the write address pointer reaches the limit of reserved address space (see chapter 8.4., Table 41, Table 42).

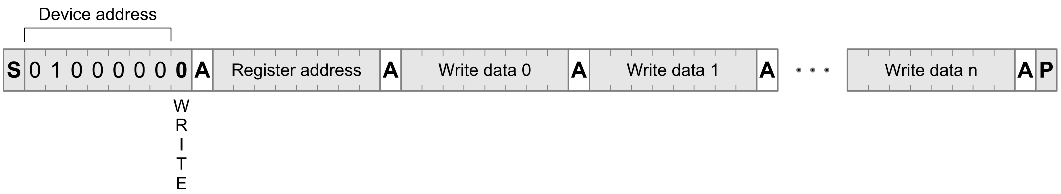



Figure 23: Multi-byte Write access through I²C

### 5.4.6. Read access

The epc660 I²C interface offers single-byte read access and multi-byte read access, where the former is basically a subset of the latter.

During a single-byte read, only one register is read. After the device address is transmitted, the master transmits the register address. After addressing the epc660 with a read-command, epc660 answers with the read data (Figure 24). The access is terminated by a not-acknowledge (NACK) and a STOP condition from the master.

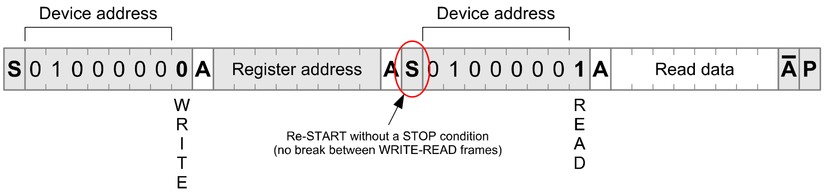



Figure 24: Single-byte Read access through I²C

During a multi-byte read operation the master transmits the device address and the address of the first register to be read. After the epc660 is addressed with a read command, epc660 answers with read data bytes until the master does not acknowledge a byte. The master is expected to terminate the access with a STOP condition thereafter (Figure 25). During the access the read address pointer is incremented epc660 internally. It is illegal to read so many data bytes that the read address pointer reaches the limit of reserved address space (see chapter 8.4., Table 41, Table 42).

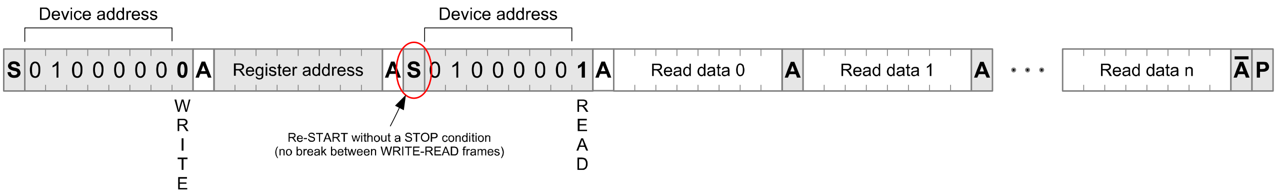



Figure 25: Multi-byte Read access through I²C

### 5.4.7. Control commands

The operating modes of the epc660 are initialized, activated, deactivated and monitored by sending several single or multi-byte write and read command sequences through I²C interface. This section lists and explains all available commands together with their access times ($f_{SCL}$ = 1MHz → $t_{SCL}$ = 1µs).

Listed in Table 14 is a summary of main control command sequences to operate the chip including the total number of command bytes transferred and their duration. There is no particular order defined for sending the commands. The only requirement is having no on-going frame acquisition process in the pipe when updating non-shadowed registers. The shadowed registers (marked with ** in the register map e.g.INTM_hi/lo, Int_len_hi/lo, etc,. see section 8.4., Table 42) can be updated on-the-fly during a frame acquisition. New values are taken into account by the next frame. The more safe approach would be finishing the last frame acquisition completely, only then update registers or change mode of operations.

| Command | Description | Length [Bytes] | Time [µs] |
|---|---|---|---|
| Reset | I²C soft reset of the chip (same effect like the $\overline{\text{RESET}}$ pin) | 2 | 20 |
| Device address activation | I2C device address activation, see chapter 5.4.4 | 2 | 20 |
| Single-byte Write | I²C Single-byte write to control registers | 3 | 29 |
| Multiple-byte Write | I²C Multiple-byte write (n bytes) to control registers | 2 + n | 20 + n x 9 |
| Single-byte Read | I²C Single-byte read from control registers | 4 | 39 |
| Multiple-byte Read | I²C Multiple-byte read (n bytes) from control registers | 3 + n | 30 + n x 9 |
| Mode set | Sine or PN mode; 4, 2, or 1 DCS set using MOD_control registers | 3 | 29 |
| Integration time (short) set | Integration time set (up to 800µs) using Int_len_hi/lo registers | 4 | 38 |
| Integration time (long) set | Integration time set (up to 26s) using INTM_hi/lo, Int_len_hi/lo registers | 6 | 56 |
| Dual Integration time (long) set | Dual int. time set using Int_len_mgx1_*, INTM_*, Int_len_* registers | 8 | 74 |
| Distance offset set | PN mode distance offset set using Dist_offset register | 3 | 29 |
| Resolution reduction set | Binning and row reduction set using Resolution_reduction register | 3 | 29 |
| ROI set | Region of interest set using ROI_* registers | 8 | 74 |
| Shutter | Soft shutter using Shutter_control registers (shutter_en bit) | 3 | 29 |
| Integration time (short) + Shutter | Int. time + soft shutter in one go! (Int_len_*, Shutter_control registers) | 5 | 47 |
| EEPROM Indirect Single Write | Indirect single write to EEPROM (bypass control registers) | 9 | 20ms |
| EEPROM Direct Burst Write | Direct byte burst write (n bytes) to EEPROM (bypass control registers) | 8 + n | n x 20ms |
| EEPROM Indirect Single Read | Indirect single read from EEPROM (bypass control registers) | 10 | 97 |
| EEPROM Direct Burst Read | Direct byte burst read (n bytes) from EEPROM (bypass control registers) | 8 + n | 78 + n x 9 |

Table 14: I²C Control commands summary

#### 5.4.8. I²C control command examples:

To simplify command sequence definitions, following C-programming language style functions are defined for the I²C master CPU:

- i2cGeneralCall(byte genAdr, byte cmd); // 20 x $t_{SCL}$ = 20µs
- i2cSingleWrite(byte devAdr, byte regAdr, byte regVal); // 29 x $t_{SCL}$ = 29µs
- i2cMultiWrite(byte devAdr, byte regAdr, byte* regVal, byte n // 20 + (n x 9 x $t_{SCL}$) = 20 + (n x 9)µs
- byte i2cSingleRead(byte devAdr, byte regAdr); // 39 x $t_{SCL}$ = 39µs
- byte* i2cMultiRead(byte devAdr, byte regAdr, byte n); // 30 + (n x 9 x $t_{SCL}$) = 30+(n x 9)µs

**Software reset with I²C general call command**

PRECONDITION: None

1. i2cGeneralCall(0x00, 0x06); // Software reset, same effect like $\overline{\text{RESET}}$ pin, 20µs
2. … // Wait for $t_{RESET}$ (> 100ns, see chapter 5.3.2., Software reset via I2C)

continue frame acquisition

**Activates device address with I2C general call command**

PRECONDITION: None

1. i2cGeneralCall(0x00, 0x04); // Activates I²C device address, 20µs
2. i2cSingleWrite(0x22, 0x92, 0x34); // Start with Sine mode, 4x DCS acquisition on I²C device 0x22...
3. i2cMultiWrite(0x22, ...
4. ...

continue frame acquisition

**Sine mode, 4x DCS mode: Acquire DCS 0 … 3 frames with integration time = 10µs**

PRECONDITION: All other registers contain default values.

1. i2cSingleWrite(0x20, 0x92, 0x34); // MOD_Control = 0x34 (mod_sel = 00, dcs_sel = 11), 29µs
2. i2cMultiWrite(0x20, 0xA2, &(0x031F), 2); // Int_len = 0x031F (Integration time = 10µs), 38µs
3. i2cSingleWrite(0x20, 0xA4, 0x01); // Shutter_Control = 0x01, (shutter_en = 1), 29µs
4. … // Acquisition starts. Wait until all 4x DCS frames are finished.

done.

**Sine mode, 4x DCS mode: Acquire DCS 0 … 3 with integration time = 10µs, followed by DCS 0 … 3 with integration time 200µs**

PRECONDITION: All other registers contain default values.

1. i2cSingleWrite(0x20, 0x92, 0x34); // MOD_Control = 0x34 (mod_sel = 00, dcs_sel = 11), 29µs
2. i2cMultiWrite(0x20, 0xA2, &(0x031F), 2); // Int_len = 0x031F (integration time = 10µs), 38µs
3. i2cSingleWrite(0x20, 0xA4, 0x01); // Shutter_Control = 0x01, (shutter_en = 1), 29µs
4. … // Acquisition starts. Wait until all 4x DCS frames are finished.
5. i2cMultiWrite(0x20, 0xA2, &(0x3E7F), 2); // Int_len = 0x3E7F (integration time = 200µs), 38µs
6. i2cSingleWrite(0x20, 0xA4, 0x01); // Shutter_Control = 0x01, (shutter_en = 1), 29µs
7. … // Acquisition starts. Wait until all 4x DCS frames are finished.

done.

**Sine mode, 2x DCS mode: Acquire DCS 0, 1 frames with integration time = 10µs**

PRECONDITION: All other registers contain default values.

1. i2cSingleWrite(0x20, 0x92, 014); // MOD_Control = 0x14 (mod_sel = 00, dcs_sel = 01), 29µs
2. i2cMultiWrite(0x20, 0xA2, &(0x031F), 2); // Int_len = 0x031F (integration time = 10µs), 38µs
3. i2cSingleWrite(0x20, 0xA4, 0x01); // Shutter_Control = 0x01, (shutter_en = 1), 29µs
4. … // Acquisition starts. Wait until all 2x DCS frames are finished.

done.

**Indirect single write to EEPROM: Store 2 bytes at address 0xE8 and 0xE9**

PRECONDITION: None

1. i2cSingleWrite(0x20, 0x11, 0xE8); // EE_ADDR = 0xE8, 29µs
2. i2cSingleWrite(0x20, 0x12, 0x22); // EE_DATA = 0x22 (Temp_tl_cal1 = 0x22), 29µs + 20ms = ~20ms
3. i2cSingleWrite(0x20, 0x12, 0x28); // EE_DATA = 0x28 (Temp_tl_cal2 = 0x28), 29µs + 20ms = ~20ms
4. ...

done.

Note 1: Start address is written in EE_ADDR.
Note 2: Each EE_DATA write starts erase/programming EEPROM. Each EEPROM write takes 20ms, then it auto-increments the EE_ADDR by 1.

Note 3: Corresponding control register value is not modified. Only EEPROM register is modified.
Note 4: EEPROM content will only be copied to corresponding control register after $\overline{\text{RESET}}$.

**Direct burst write to EEPROM: Store 8 bytes at address 0xE8 ... 0xEF**

PRECONDITION: None

1. i2cMultiWrite(0x20, 0xE8, &(0x22,0x28,0x20,0x26,0x18,0x17,0x19,0x19), 8); // Store 8x bytes,
   // calibration point registers Temp_xx_cal1/2, 20µs + 8 x (9µs + 20ms) = ~160.1ms
2. i2cSingleWrite(0x20, 0x02, 0x05); // MEM_CTRL_CONF = 0x05 (ee_direct_access = 0), 29µs

done.

Note 1: Start address is sent only one time.
Note 2: Corresponding control register value is not modified. Only EEPROM register is modified.
Note 3: EEPROM content will only be copied to corresponding control register after $\overline{\text{RESET}}$.

**Indirect single read from EEPROM: Read 2 bytes from address 0xEA and 0xEB**

PRECONDITION: None

1. i2cSingleWrite(0x20, 0x11, 0xEA); // EE_ADDR = 0xEA, 29µs
2. cal1 = i2cSingleRead(0x20, 0x12); // cal1 = EE_DATA (Temp_tr_cal1 = 0x20), 39µs
3. cal2 = i2cSingleRead(0x20, 0x12); // cal2 = EE_DATA (Temp_tr_cal2 = 0x26), 39µs
4. ...

done.

Note 1: Start address is written in EE_ADDR.
Note 2: Each EE_DATA read auto-increments the EE_ADDR by 1.
Note 3: Corresponding control register value is not modified. Only EEPROM is read.

**Direct burst read from EEPROM: Read 8 bytes from address 0xE8 ... 0xEF**

PRECONDITION: None

1. i2cSingleWrite(0x20, 0x02, 0x07); // MEM_CTRL_CONF = 0x07 (ee_direct_access = 1), 29µs
2. cal_arr[] = i2cMultiRead(0x20, 0xE8, 8); // Read 8 bytes calibration points Temp_xx_cal1/2, 20µs + 8 x 9µs = 92µs
3. i2cSingleWrite(0x20, 0x02, 0x05); // MEM_CTRL_CONF = 0x05 (ee_direct_access = 0), 29µs

done.

Note 1: Start address is sent only one time.
Note 2: Corresponding control register value is not modified. Only EEPROM is read.

### 5.5. LED driver and DLL

#### 5.5.1. LED driver

The LED driver is an open-drain MOSFET device. It can sink up to 350mA peak current during operation. The modulation signal is a square-wave up to typically 20MHz with a duty cycle of 50% which depends on the mode of operation.

**IMPORTANT:** There are non-modulating DC modes (e.g. black & white with LED/LD illumination) which keeps the LED driver always turned on. In this case, the user has to take care that LED driver and the epc660 chip does not exceed the maximum operating limits.

The LED_driver register contains several static bit fields that can be used for setting drive strength, polarity, etc. depending on the external LED/LD circuitry used in the application (see chapter 8.7.6.). These bit fields must not be modified during the frame acquisition.

- The led_drv_sel bits set LED driver output maximum current e.g. for 200mA to 100% , 64%, 46% or 33%.
- The led_on bit permanently turns on the LED/LD (torch function). Keep care to the power dissipations if you operate the LEDs and laser diodes in this mode.
- The led_drv_en bit enables the internal LED driver circuits. The enabled after reset (default), can be disabled if there will be no frame acquisition for a long time.
- The led_inv_en bit sets the polarity of the LED driver. This must be set/cleared with respect to external LED/LD driver circuit topology and stored in the EEPROM. It is cleared after reset (default), which matches to the simple external LED circuit shown in Figure 14
- led_ssr_en bit sets a slower slew rate for the LED driver (part of the LED driver transistor is switched on with a delay to lower the current peaks).

#### 5.5.2. DLL (Delay Locked Loop)

The LED driver and external LED's/LD's phase stability suffer mainly from temperature, ageing, etc. of the system. This shows itself as distance error. The built-in DLL circuit minimizes this behaviour. It minimizes the above mentioned long term deviations in the phase delay of critical signals. They are from seconds up to several minutes and longer and impact the performance of the distance accuracy. The DLL can be used with Sine and PN modes. It is located between the modulator and the LED driver (see Figure 27).

The modulator invokes the DLL before DCS frame acquisition at the rate set by the DLL_measurement_rate register. Otherwise the DLL period is skipped and the last compensation delay value is preserved on the led_mod signal path. Figure 26 shows the example for 4x DCS frames and DLL_measurement_rate = 4.

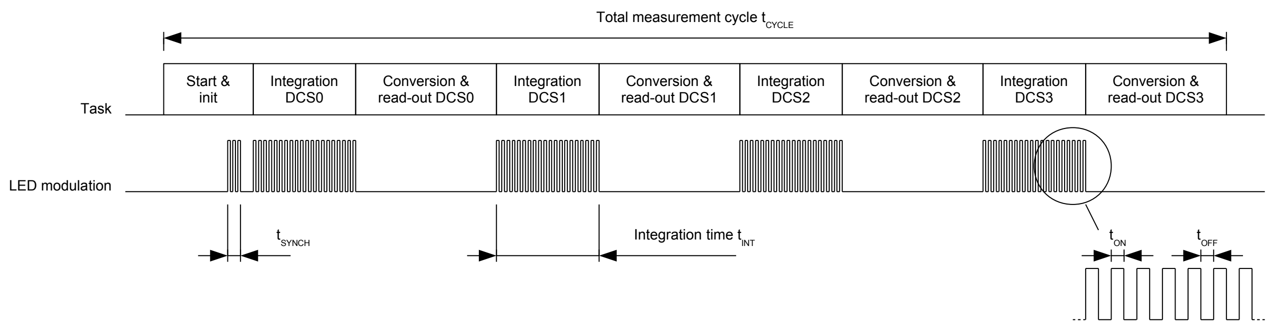



Figure 26: DLL synchronization example for 4x DCS frames

Is the DLL invoked, the mga demodulation signal, originated from the modulator, goes first through digital delay stages. Secondly, it is distributed to the individual pixels in the pixel field (red path, until point A).

The same time, the led_mod signal goes through the DLL, the LED driver and comes to the LED pin. This pin drives the external LED circuit on the PCB, which generates the modulated IR light (blue path, until point B).

These two signal paths are subject to phase delay deviations depending of e.g. temperature or aging. This delays translate into distance errors. The DLL minimizes deviations by adding a certain delay on the led_mod signal (x-Delay).

The DLL compares the demodulation signal phase coming from a pixel as mga (point A) with the current feedback signal from the LEDFB pin of the external LED circuit (point B). The distance reading error of the epc660 is minimal if the delays to the signals from the points A and B are identical.

User should minimize the path delay from the feedback node between the resistor dividers to the LEDFB pin (yellow). This can be achieved by placing the resistor dividers as close as possible to the LEDFB pin on the PCB. Other noisy circuits should be placed away from the LEDFB pin and resistor dividers.

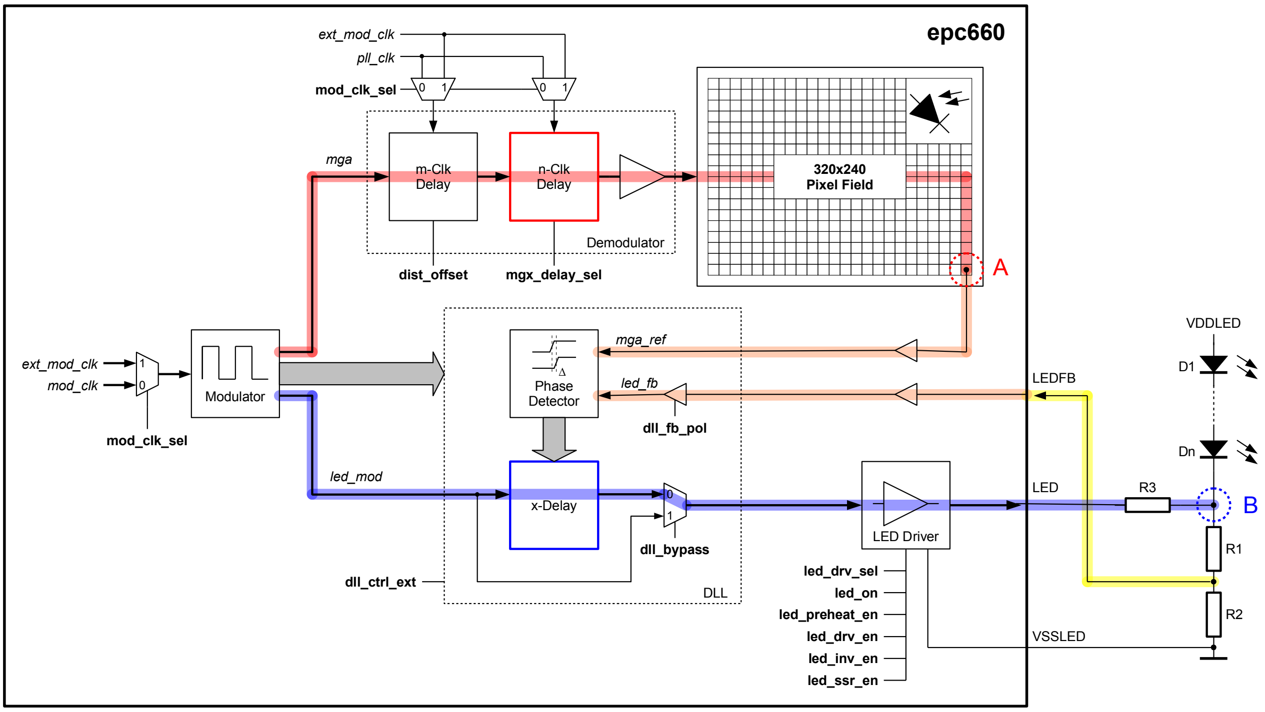



Figure 27: Simplified modulator, demodulator, DLL and LED driver diagram

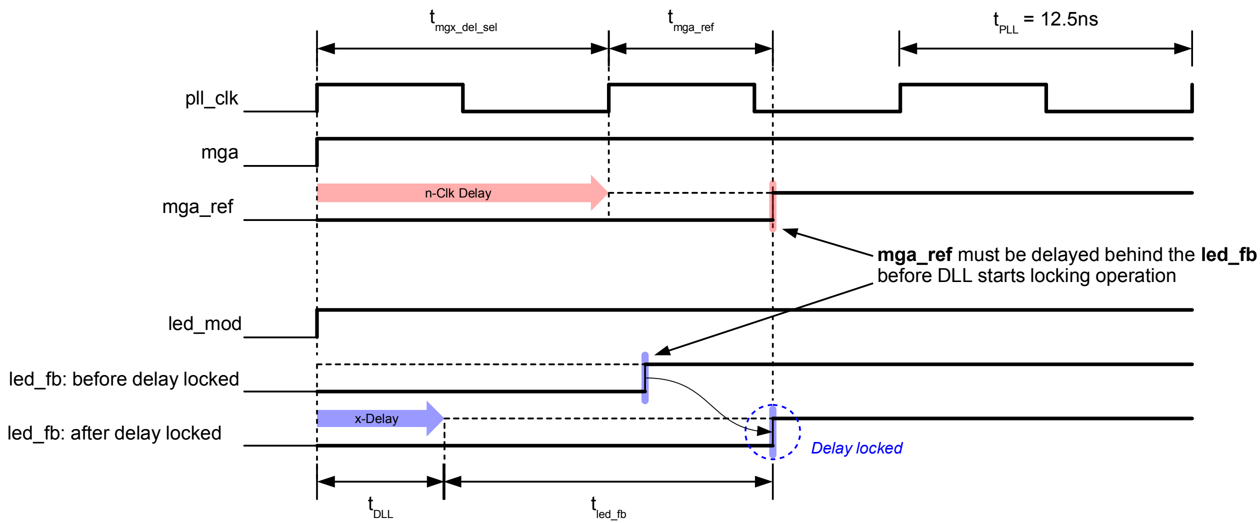



Figure 28: DLL delay locking operation

### DLL locking operation

Before the DLL starts its locking operation, the mga_ref signal must be delayed behind the led_fb signal (see Figure 28). This is achieved by setting mgx_delay_sel = 1 or higher in the register Demodulation_delays, which gives an initial 12.5ns phase offset to the mga_ref signal (PLL is set to 80MHz). Next, the modulator starts toggling the mgx and led_mod signals for a certain time without enabling the DLL (DLL pre-synchronization: DLL_en_del). Finally, the modulator enables the DLL and keep toggling the mgx and led_mod signals until the locking operation is finished (DLL_en). The DLL pre-synchronization and DLL lock time is set to 10µs, respectively 30µs, in Figure 26 indicated as $t_{SYNC}$. These default values can be changed by DLL_en_del/DLL_en registers (see Table 42 , DLL Synchronization registers).

When locked, the DLL adds and freezes the compensation delay (x-Delay) to bring point A and B together in time, until the same operation is restarted by the modulator again. The rate of DLL measurements is set in terms of number of DCS frames by DLL_measurement_rate register. Note, this value must be set to multiples of number of DCS frames selected by the MOD_Control.dcs_sel register e.g. when dcs_sel = 11, the system is in the 4x DCS mode, then set DLL_measurement_rate = 4. This way, all DCS frames (0 … 4) are acquired with the same DLL compensation delay and the samples stay correlated with each other. For DLL_measurement_rate = 0, last measured compensation delay value is kept and used for the next DCS frames.

**IMPORTANT:** The DLL has a delay locking range of 6ns ... 76ns with 7ps fine delay steps. There is always a minimum 6ns delay inserted on the led_mod line when the DLL is enabled. The LED driver, the external LED/LD circuit and the LEDFB input amplifier delays are added on top. Therefore, mgx_del_sel must always set to 1 or higher.

If the dependent delay variation of the external LED/LD circuit is more than the one pll_clk period, mgx_del_sel must set to a higher value the way, that the DLL can compensate this wide-range variation. LED/LD circuits placed on the board far away from the epc660 or placed even more on another PCB are connected with long wires. This can cause long time offset.

### 5.6. Pixel Field

#### 5.6.1. Pixel coordinates

The epc660 pixel field consists of a total of 328 x 252 pixels. 6 rows top/bottom and 4 columns left/right on the periphery of the pixel field contain dummy pixels which cannot be used for 3D-TOF and black & white imaging. The active pixels, so called “effective pixels”, can be used for TOF and black & white imaging modes. They are in the central rectangle area forming a full QVGA resolution 320 x 240 pixel (Figure 29).

The upper-left corner (top view on chip) is the origin (0/0) of the epc660 pixel coordinate system. Pixel x-axis starts at 0 and counts up to 327 to the right. Pixel y-axis starts at 0 and counts up to 251 to the bottom. All readout modes (ROI, resolution reduction, binning, etc.) and control registers use this coordinate system to set or change modes of the chip.

User should not readout dummy columns (x = 0 … 3; 324 … 327) or dummy rows (y = 0 … 5; 246 … 251) as there cannot be acquired useful imaging data from these pixels.

The QVGA rectangle with the effective pixels range between the upper-left (4/6) and bottom-right (323/245) pixel. It is set as the default ROI area during start-up.

The pixel field is split vertically into two equal parts as pixel field top and bottom. The data read-outs are duplicated symmetrically on the top and bottom of the chip in order to double the frame rate of the epc660.

Default readout starts in the middle of the row axis and continues symmetrically towards up and down, row by row and in parallel, to the top and bottom of the pixel field. As example, default readout starts with rows R125/R126, followed by R124/R127, and so on. The vertical readout direction can be changed by the application appropriately.

The internal readout of a row is split in two sections: first all even pixels; second all odd pixels. Later on the TCMI interface presents the row in the regular order with even and odd pixels mixed.

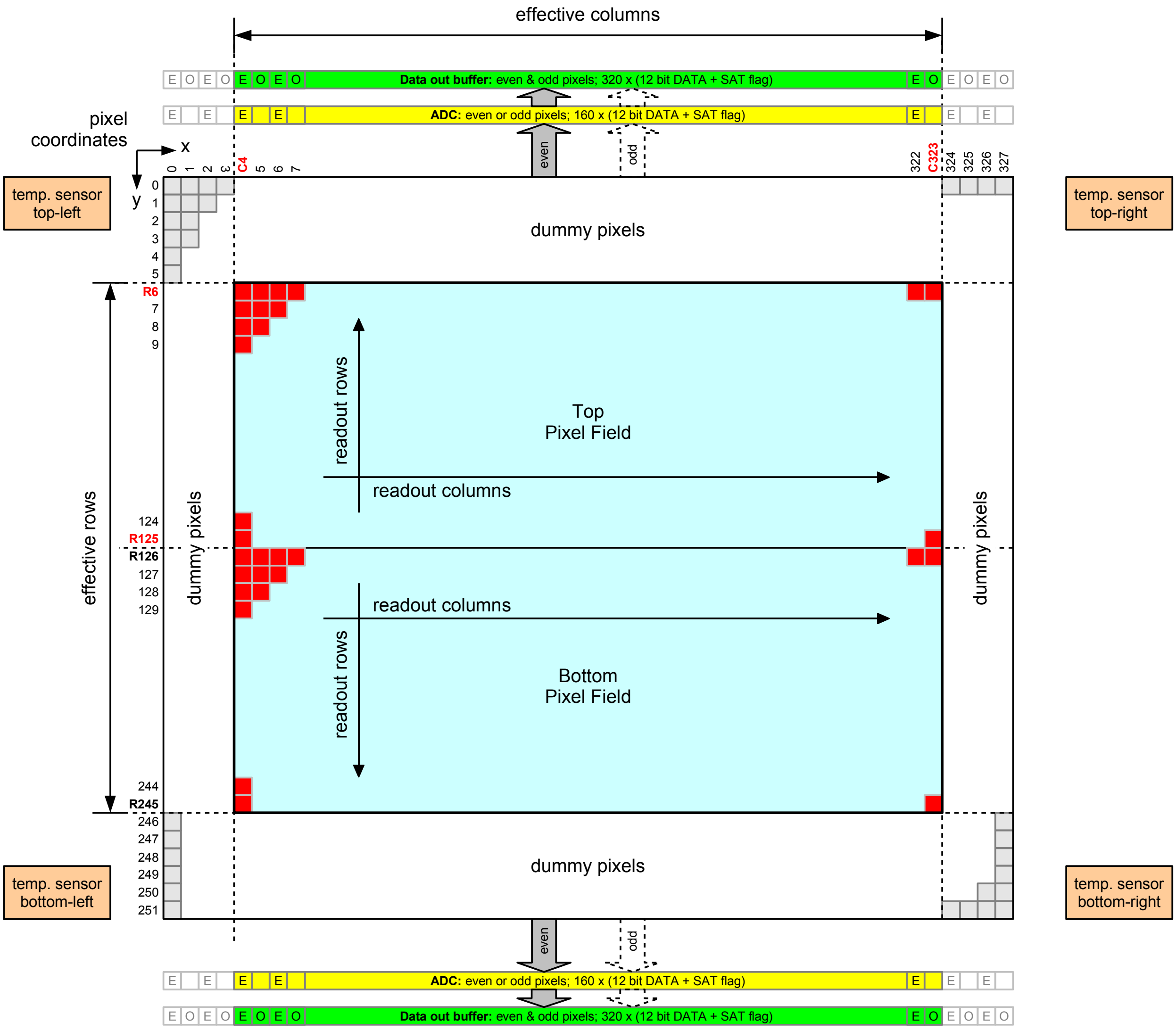



Figure 29: Pixel field coordinates with row and column numbering scheme (top-view, solder balls are bottom side)

### 5.6.2. Pixel architecture

The pixels are placed in groups 2x2 pixels, called herein “pixel group”. The pixel group performs two basic operations: Measurement (integration) and readout (ADC). Pixels are named as UE (Upper-row, Even-column), UO (Upper-row, Odd-column), LE (Lower-row, Even-column) and LO (Lower-row, Odd-column) depending on their location within the pixel group (see Figure 29). Pixels with the same name are controlled simultaneously in the whole pixel field. More precisely, pixels in the upper and lower rows are controlled simultaneously during measurement, pixels in the even and odd columns are controlled simultaneously during readout.

The pixel group architecture allows the epc660 to operate the pixel field in different modes and in combinations thereof according the following chapters.

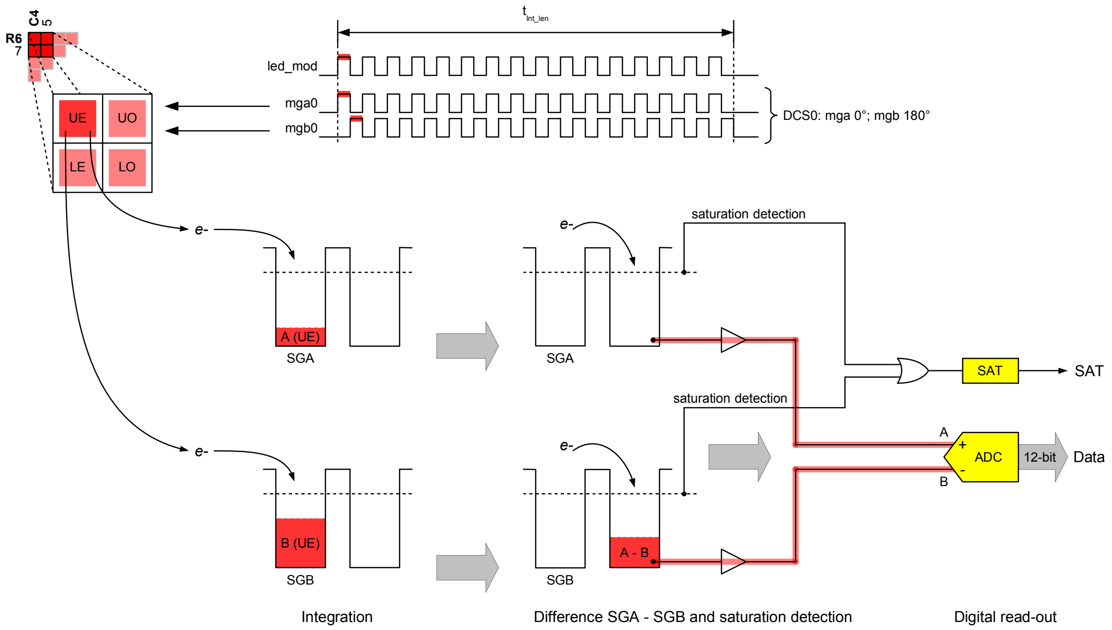



Figure 30: The 2x2 pixel group and the simplified function overview

Each pixel has its own pair of storage gates SGA and SGB. During the integration time, they accumulate the charges (e-) created by the reflected modulated light coming from the object (see section 6., Measurement Modes). They are controlled by the mga and mgb demodulation signals. After the measurement is finished, the readout phase starts. The charges stored in the storage gates SGA and SGB are read out as a difference A – B (ambient-light suppression) and converted into a single 12-bit digital value and a 1 bit saturation flag.

Depending of the operation mode the ADC readout is different:

| | |
|---|---|
| Differential mode: | Used for 3D TOF sine and PN. It shows the difference of the two storage gate charges.<br>Digital output values are signed values. |
| Single-ended mode: | Used for black & white images. It shows the charge of the selected storage gate.<br>For the SGA readout the conversion result is a positive 11-bit value (except noise).<br>For the SGB readout the conversion result is a negative 11-bit value (except noise). |

**IMPORTANT:** The application software must take care about the negative (-) single ended sample value, accordingly.

**Single MGX mode (default)**

When dual_mgx_mode = 0, all UE, UO, LE, LO storage gates work simultaneously during measurement operation. The storage gate control signals mga, mgb that are coming to all pixels are driven identically. In this mode, one DCS frame (0 …3) is acquired at the time (see Figure 31).

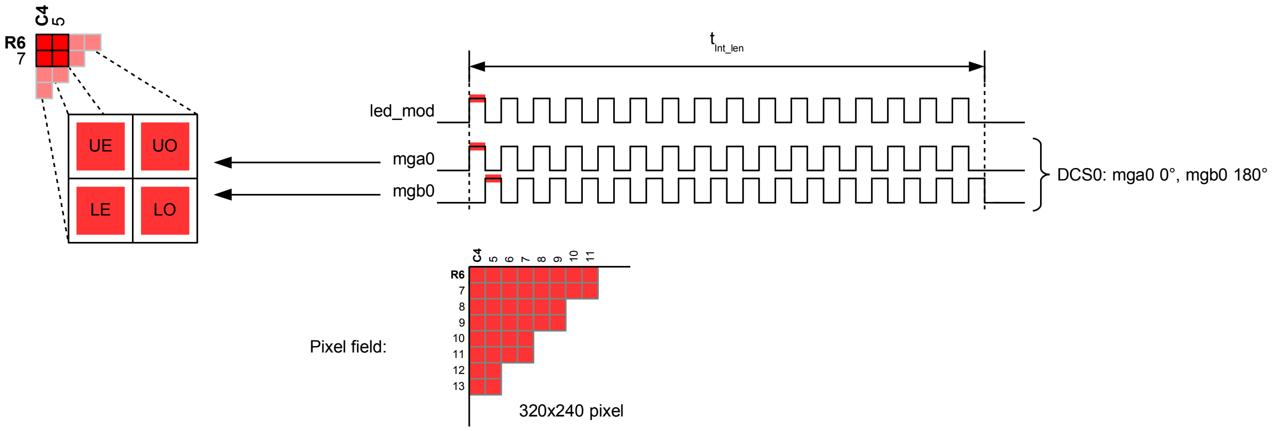



Figure 31: Single MGX mode: upper and lower pixel rows are controlled identically with mgx0

**Dual MGX mode with same integration time (motion blur reduction)**

When dual_mgx_mode = 1 and int_len_mgx1_en = 0, the UE, UO, LE, LO storage gate control signals mgx (mga, mgb) are generated as two independent groups as mgx0 (mga0, mgb0) and mgx1 (mga1, mgb1). As a result, two correlation frames DCS0 and DCS1 (or DCS2 and DCS3) can be acquired simultaneously (see Figure 32). . The upper row pixels store DCS0 (or DCS2) while the lower row pixels store DCS1 (or DCS3). The vertical pixel pairs (e.g. UE/LE) must be treated for distance calculation as if they are one single pixel. This comes at the cost of a reduced resolution along the y-axis. The result provides a total of 320x240 pixel field readout with an effective 3D TOF resolution of 320x120 pixel.

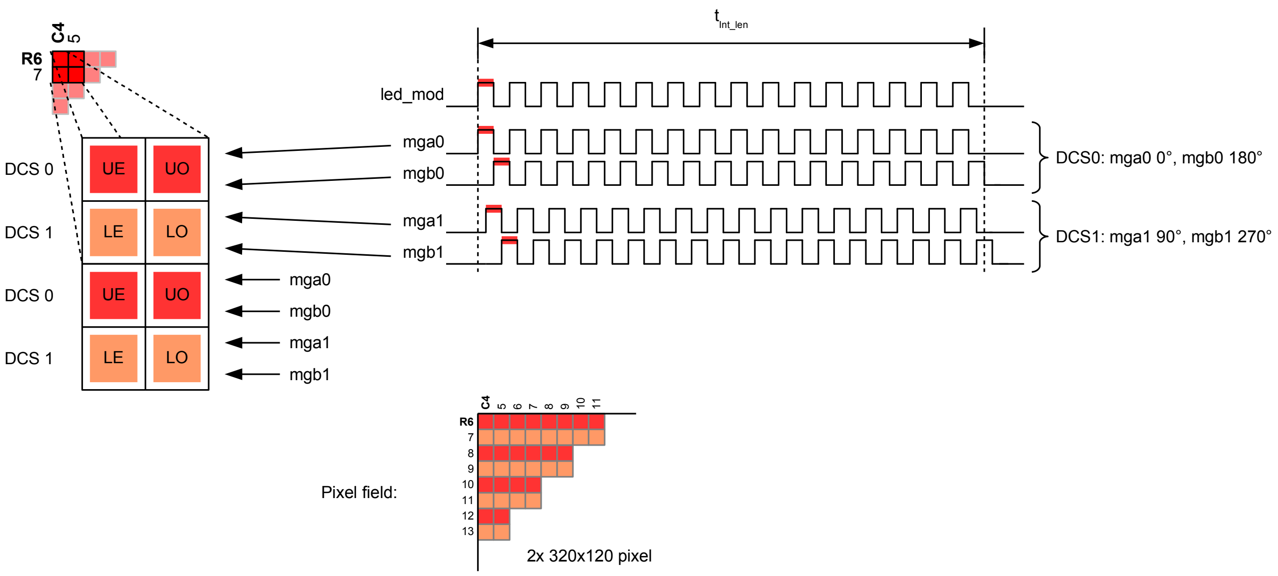



Figure 32: Dual MGX mode with same Integration time
(upper and lower rows independently controlled by mgx0 and mgx1 with different phase shifts)

**IMPORTANT:** This mode requires an optical anti-aliasing filter (similar to a colour imager), so that adjacent pixels receive the same light. Otherwise, light from different locations cause 'zebra like' artefacts at the end of every strip.

**Dual MGX mode with different integration times (High dynamic range)**

When dual_mgx_mode = 1 and int_len_mgx1_en = 1, the UE, UO, LE, LO storage gate control signals mgx (mga, mgb) are generated as two independent groups as mgx0 (mga0, mgb0) and mgx1 (mga1, mgb1). Both groups provide exactly the same DCS modulation signals (phases). One stops earlier than the other due to different integration times (see Figure 33). The effect: As a consequence, the two pixels collect different amount of light simultaneously. There is no restriction about which integration time is shorter or longer with respect to the other. The upper row pixels integrate with Int_len_hi/lo register value, while the lower row pixels integrate with Int_len_mgx1_hi/lo value. The upper and lower pixels (e.g. UE, LE) can be used independently for distance calculation. This comes at the cost of a reduced resolution along the y-axis. Instead of one frame with 320x240 pixels, a single readout provides two DCS or black and white frames with an effective resolution of 320x120 pixels but with different integration times.

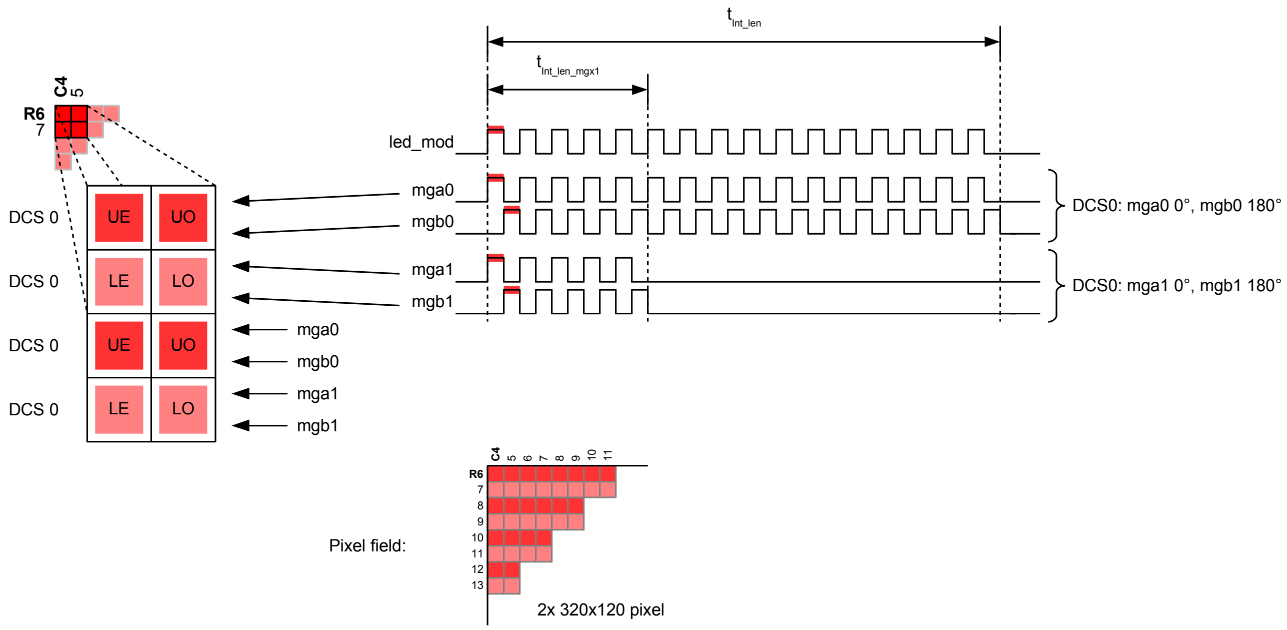


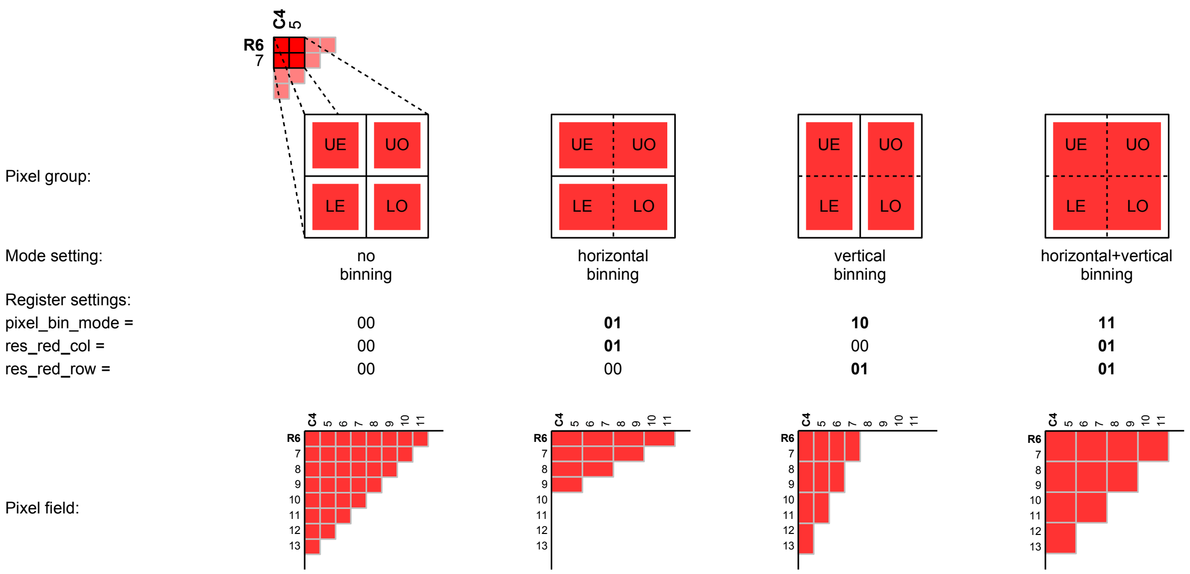

Figure 33: Dual MGX mode with different integration times
(upper and lower rows independently controlled by mgx0 and mgx1. One stops earlier than the other)

**Pixel binning**

The charges accumulated in the storage gates during measurement can be combined in various ways during the readout phase. This is called binning. There are three binning choices: horizontal, vertical, horizontal+vertical (see Figure 34). Advantages of binning can be: higher sensitivity, reduced integration time, better ratio signal to shot-noise and faster readout of frames (Note: TCMI data rate stays unchanged).

Binning modes require corresponding resolution reduction modes being enabled at the same time. Refer to Pixel operating and readout control registers, Resolution reduction, Table 42.

**IMPORTANT:** Pixel binning cannot be used with dual MGX modes.




| | | | | |
|---|---|---|---|---|
| Mode setting: | no binning | horizontal binning | vertical binning | horizontal+vertical binning |
| Register settings: | | | | |
| pixel_bin_mode = | 00 | **01** | **10** | **11** |
| res_red_col = | 00 | **01** | 00 | **01** |
| res_red_row = | 00 | 00 | **01** | **01** |
| Reduction of pixel field resolution: | 320 x 240 | 160 x 240 | 320 x 120 | 160 x 120 |
| Increase of sensitivity: | 1 x | 2 x | 2 x | 4 x |
| Increase of ratio signal to shot-noise: | 1 x | $\sqrt{2}$ x | $\sqrt{2}$ x | 2 x |
| Decrease of DCS frame readout time: | 1 x | 1/2 x | 1/2 x | 1/4 x |
| Increase of DCS frame rate: | 1 x | 2 x | 2 x | 4 x |

Figure 34: Overview pixel binning modes and readout

### 5.6.3. Pixel readout modes

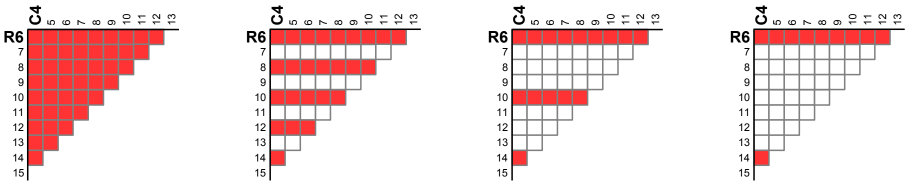
**Resolution Reduction**

The aim of resolution reduction is to shrink the data set to the really necessary amount of data required for for the application. The advantage is the reduced amount of data which has to be processed online for the final measurement result (reduced frame buffers, processing time and so on), shorter readout times for frames with short integration times and an increase of the achievable data rate.

Resolution reduction can be performed in terms of columns and rows, independently. Reductions by reading only every 2nd column on x-axis and every 2nd , 4th and 8th row on y-axis are supported. The resolution reduction can be combined with pixel binning modes (see chapter before Pixel binning), ROI (see next section) and Dual MGX modes (see Figure 35 - Figure 38 for valid combinations).

Note: Resolution reduction along x-axis is supported for every 2nd column only. This is required by the horizontal binning mode (see chapter Pixel binning). Reduction to 4th and 8th column only are not supported as they are not providing any frame rate improvement.

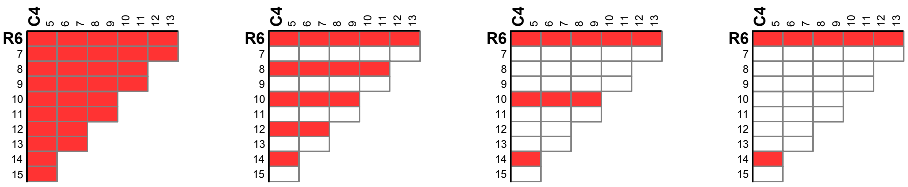
| Register settings: | | | | |
|---|---|---|---|---|
| pixel_bin_mode = | 00 | 00 | 00 | 00 |
| res_red_col = | 00 | 00 | 00 | 00 |
| res_red_row = | 00 | **01** | **10** | **11** |




| | | | | |
|---|---|---|---|---|
| Pixel field resolution: | 320 x 240 | 320 x 120 | 320 x 60 | 320 x 30 |
| Frame rate: | 1x | 2x | 4x | 8x |

Figure 35: Resolution reduction of rows (y-axis) without binning

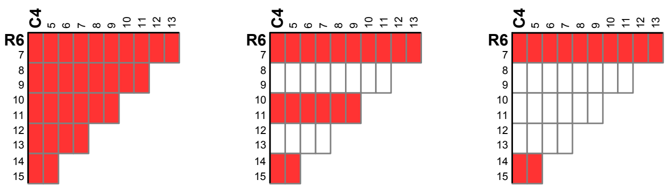
| Register settings: | | | | |
|---|---|---|---|---|
| pixel_bin_mode = | **01** | **01** | **01** | **01** |
| res_red_col = | **01** | **01** | **01** | **01** |
| res_red_row = | 00 | **01** | **10** | **11** |




| | | | | |
|---|---|---|---|---|
| Pixel field resolution: | 160 x 240 | 160 x 120 | 160 x 60 | 160 x 30 |
| Frame rate: | 2x | 4x | 8x | 16x |

Figure 36: Resolution reduction of rows (y-axis) combined with horizontal binning

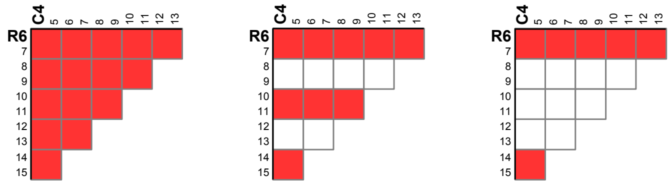
| Register settings: | | | |
|---|---|---|---|
| pixel_bin_mode = | **10** | **10** | **10** |
| res_red_col = | 00 | 00 | 00 |
| res_red_row = | **01** | **10** | **11** |




| | | | |
|---|---|---|---|
| Pixel field resolution: | 320 x 120 | 320 x 60 | 320 x 30 |
| Frame rate: | 2x | 4x | 8x |

Figure 37: Resolution reduction of rows (y-axis) combined with vertical binning

| Register settings: | | | |
|---|---|---|---|
| pixel_bin_mode = | **11** | **11** | **11** |
| res_red_col = | **01** | **01** | **01** |
| res_red_row = | **01** | **10** | **11** |



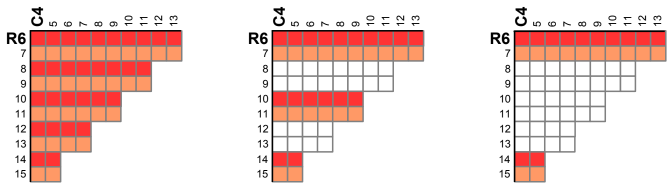
| | | | |
|---|---|---|---|
| Pixel field resolution: | 160 x 120 | 160 x 60 | 160 x 30 |
| Frame rate: | 4x | 8x | 16x |

Figure 38: Resolution reduction of rows (y-axis) combined with horizontal and vertical binning

The resolution reduction modes can be mixed also with Dual MGX modes, either with Dual MGX mode with same integration time (motion blur reduction, see Figure 39) or with MGX mode with different integration times (high dynamic range, see Figure 40). For details of these two modes, refer to chapter 5.6.2, Pixel architecture.

**IMPORTANT:** Pixel binning cannot be used with dual MGX modes.

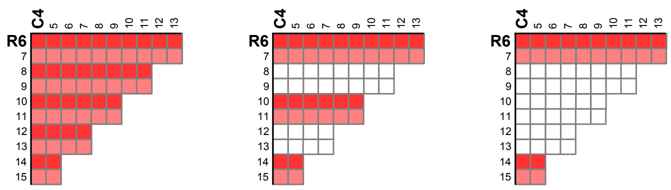
| Register settings: | | | |
|---|---|---|---|
| dual_mgx_mode = | **1** | **1** | **1** |
| pixel_bin_mode = | 00 | 00 | 00 |
| res_red_col = | 00 | 00 | 00 |
| res_red_row = | 00 | **01** | **10** |



| | | | |
|---|---|---|---|
| Pixel field resolution: | 2 x 320 x 120 | 2 x 320 x 60 | 2 x 320 x 30 |
| Frame rate: | 1x | 2x | 4x |

Figure 39: Resolution reduction of rows (y-axis) combined with dual MGX mode
with same integration time (motion blur reduction)

| Register settings: | | | |
|---|---|---|---|
| dual_mgx_mode = | **1** | **1** | **1** |
| pixel_bin_mode = | 00 | 00 | 00 |
| res_red_col = | 00 | 00 | 00 |
| res_red_row = | 00 | **01** | **10** |



| | | | |
|---|---|---|---|
| Pixel field resolution: | 2 x 320 x 120 | 2 x 320 x 60 | 2 x 320 x 30 |
| Frame rate: | 1x | 2x | 4x |

Figure 40: Resolution reduction of rows (y-axis) combined with dual MGX mode
with different integration times (high dynamic range)

**ROI (Region of Interest)**

The ROI feature allows to readout and transfer the portion of the pixel field data which is necessary for the application. The advantages are (same as for the resolution reduction) the reduced amount of data which have to be processed online for the final measurement result (Reduced data readout time, frame buffers, processing time and so on), also shorter readout times for frames with short integration times and so far an increase of the possible data rate. For integration times much shorter than the row conversion time (in the µs range, see Figure 42), the frame rate directly scales with the number of rows readout via ROI setting.

ROI is always operational and cannot be disabled. It works symmetrically over the top and bottom pixel fields. Therefore it must be set only in the top pixel field between the minimum top-left [C4,R6] and the maximum bottom-right [C323,R125] pixel coordinates (see registers ROI_tl_x_hi/lo, ROI_tl_y, ROI_br_x_hi/lo, ROI_br_y in Table 42, Frame column / row mapping registers). The symmetric part in the bottom pixel field is generated simultaneously and always.

The ROI must be set by the application always starting with even row and column and ending with odd row and column. Top-left coordinates must be smaller than the bottom-right.

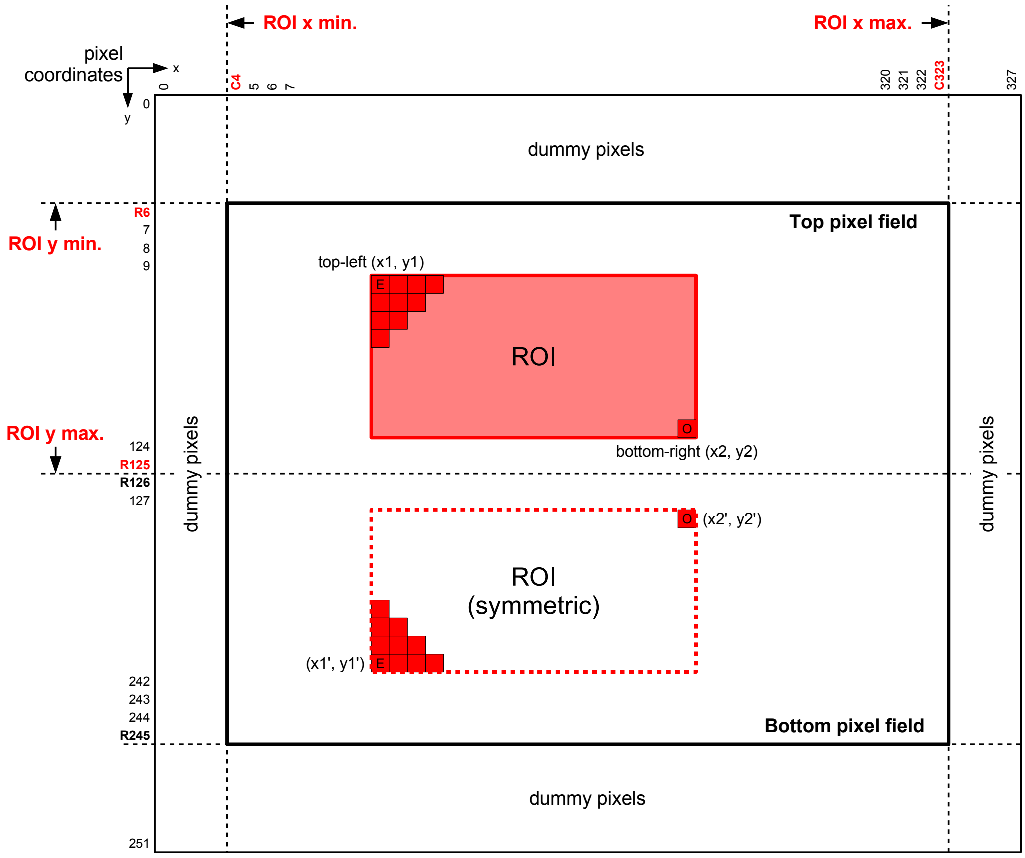



Figure 41: Region of interest (ROI)

The ROI registers can be changed on-the-fly via I²C without waiting the completion of current frame's readout cycle. The new values will be used immediately with the next frame. In parallel, the application must make sure that the frame buffers of the application are synchronized/readjusted to the new ROI frame size prior to data readout starts.

**IMPORTANT:**

1. ROI can be set to a minimum rectangle of columns by rows of 6 by 2.
2. If row reduction is enabled, the minimum number of ROI rows is inversely scaled, e.g.: **row reduction by 2** makes the minimum ROI to 6 by **4**.
3. If column reduction is enabled, the minimum number of ROI columns is inversely scaled, e.g.: **column reduction by 2** makes the minimum ROI to **12** by 2.

Note: For the emulation of the epc635 refer to the application section: chapter 7.6.4., epc635 emulation.

#### 5.6.4. Pixel saturation detection

The pixels collect continuously modulated and non-modulated, ambient (background) light during the integration period. Depending on these light intensities, sometimes the pixels collect more charge (over-exposure) than they can accommodate in their storage gates (refer to Figure 30). In such a case, the 12 bit sample data is not valid and cannot be used. Therefore, each pixel generates a “saturation detection” flag along with the sample data, so that the data can be discarded by the application.

The saturation flag is generated in the pixel binning modes as well. It is set when one or more storage gates of the binned pixels is/are saturated or the sum thereof. Due to this architecture, two binned pixels cannot collect twice the full storage gate charge.

The saturation detection flag is transmitted via XSYNC_SAT_CFG pin synchronously to the DATA[11:0] for every 12 bit pixel sample. If XSYNC_SAT_CFG pin is programmed for an another function by setting tcmi_xsync_sat_sel = 0 or not connected at all, tcmi_data_sat_en = 1 can optionally drive all DATA[11:0] pins to 0xFFF when the pixel is saturated (see TCMI_polarity register in chapter 8.4., Table 42, TCMI registers). In this configuration, the application must discard those samples equal to 0xFFF, otherwise use it for calculation.

#### 5.6.5. Ambient-light suppression

An important function of the 3D TOF pixel is the ability to separate the self-emitted and reflected modulated light from the ambient light. The built-in ambient-light suppression removes the DC or low frequent signal distortions, caused by foreign light sources e.g. sunlight, daylight, room illumination, etc., from the measuring signal. The user has not to take care of this, it is done by the pixel automatically by using only the charge difference of the storage gates (see Figure 30). To see the capability of this function, refer to Table 7 for example values as a function of the wavelength and compared to sunlight. Similar to the system's sensitivity of the modulated light, is the ambient-light suppression also a function of the integration time. The longer the integration time, the more the measurement becomes sensitive to the ambient-light.

Notes:

- The ambient-light suppression of the chip must not be confused with the ambient-light measurement. It is a fixed built-in functionality, which is removing the DC light component from the AC measurement signal only.
- A DC or AC photo signal can be generated by ambient-light (e.g. sunlight) or by cross-talk from the IR-LEDs. However, if this is above the stated maximum value, then the sensor or the input electronics are saturated. This blocks the detection of the AC modulation signal.

### 5.7. Temperature Sensors

There are four temperature sensors located at the corners of the pixel field (Figure 29). They are not calibrated and reading the values as they are. Temperature sensor registers can be read on-the-fly via I²C interface while a frame acquisition is going on. Internal synchronization guarantees that always a valid temperature value is readout via I²C as long the following conditions are fulfilled:

Each sensor is sampled synchronously to the row readout cycles. Always four consecutive readings are summed up to the resulting 14 bit value written into the corresponding Sum_Temp_xx_hi/lo registers (see Table 41, Temperature sensor registers). The update of the upper or lower temperature sensor values takes place every time, the corresponding pixel field half (top or bottom) has been row readout two times. Temperature sensors will be reset with every new frame start.

There will be not enough information to calculate the temperature value

- after a frame start until two row readout cycles are performed per half pixel field.
- or if the ROI and the binning are set such that less than two row readout cycles are performed per half pixel field and per frame.

### 5.8. TOF camera interface (TCMI)

The TCMI is a programmable high-speed parallel data output interface to off-load raw DCS (Differential Correlation Samples) data for all or part of the 320x240 pixel field (see chapter 5.6.) to an external application, e.g. to a DSP processor. It is programmable through the TCMI registers.

The TCMI interface provides:

- programmable active high/low logic levels on all clock and control signals individually (default: all active high).
- programmable output clock rates.
- 12-bit parallel output with 1 bit saturation flag.
- programmable 'ITU-R 656 like' synchronization options which allows easy connection to a wide variety of high-performance low-cost SoC embedded processors available in the market (supports hardware synchronization, but not embedded 0xFF/0x00 sync data packets)

When the integration period is terminated and all ADC conversions are finished, the readout results are moved into the data out buffers to be immediately transmitted via the TCMI interface to the outside of the chip. The parallel conversion of two full rows (top and bottom pixel field) takes in total 31.25µs independent of the number of selected columns.

Depending of the mode selection (4x DCS, 2x DCS, ...) a programmable number of DCS frames are generated by the epc660. The generated data is streamed out as a complete block of 1 DCS frame, one after the other following the procedure described in chapter 5.6. Each row contains 12-bit DCS values and the SAT bit for the selected number of pixels per row.

The pixel values are streamed out as 12 bit signed numbers. Two rows are streamed out in sequence together, the first one from the top and the second one from the bottom pixel field e.g. R125 (C4, C5, … C323), R126 (C4, C5, … C323), R124 (C4, C5, … C323), R127 (C4, C5, … C323), R6 (C4, C5, … C323), R245 (C4, C5, … C323) and so on. The stream-out of a row pair takes 16 µs with default clock settings (40MHz TCMI clock rate).

Transmitting columns left to right or vice-versa can be changed by setting the bit col_dir_sel (see Readout_dir **, Table 42). The row readout direction cannot be changed.

The transfer of a DCS frame should not be interrupted or stopped, once it is started. The application should have enough bandwidth to receive all transmitted frames.

#### 5.8.1. TCMI clock mode

The TCMI interface supports the continuous clock mode. It uses the internal tcmi_clk as a time base. The user must program the appropriate clock speed before running the TCMI interface. It is the output data transfer rate (refer to signal DCLK).

The frames are transmitted at high-speed using all *SYNC (VSYNC_A0, HSYNC_A1, XSYNC_SAT_CFG), DATA[11:0] and DCLK outputs (Figure 42).

The DCLK signal toggles continuously. The frequency is programmable to 10, 20, 40, 80MHz. The user application is responsible to set the correct clock rate in order to achieve the most efficient data acquisition throughput for the target application. The choice depends on user system factors such as e.g. application CPU/DSP's GPIO and/or bus speeds.

| tcmi_clk_div (0x89) | | |
|---|---|---|
| **20MHz DCLK** | **40MHz DCLK** | **80MHz DCLK** |
| **[#]** | **[#]** | **[#]** |
| **2** | **1** [1] | **0** |

Table 15: DCLK settings

[1] Note: Default setting 40MHz TCMI clock.

All *SYNC*, DATA[11:0] signals are synchronously updated with positive edge of the DCLK signal when its polarity is set as active-high; with negative edge of the DCLK signal when its polarity is set as active-low. The non-active edge of the DCLK output can be used by the receiving end (application CPU) as a sampling clock. It should approximately be in the center of the data.

By using the default configuration, the active states of VSYNC_A0 and HSYNC_A1 signals indicate blanking periods during the frame transmission. It is a legacy feature from the ITU-R 656 standard for video signal transmission. While DCLK toggles continuously, any data during the blanking periods are not valid and must must be ignored.

As soon as the measurement result of the first row of the new frame is available, VSYNC_A0 and HSYNC_A1 are set consecutively with the next active edge of DCLK. VSYNC_A0 is active from the start until the end of the each complete frame. Whereas, HSYNC_A1 indicates the validity of the DATA[11:0] and XSYNC_SAT_CFG (saturation bit) from the start until the end of a row pair.

By default, the XSYNC_SAT_CFG pin is used for the saturation bit. Optionally, it can be programmed to indicate the end of a frame (tcmi_xsync_sat_sel = 0).

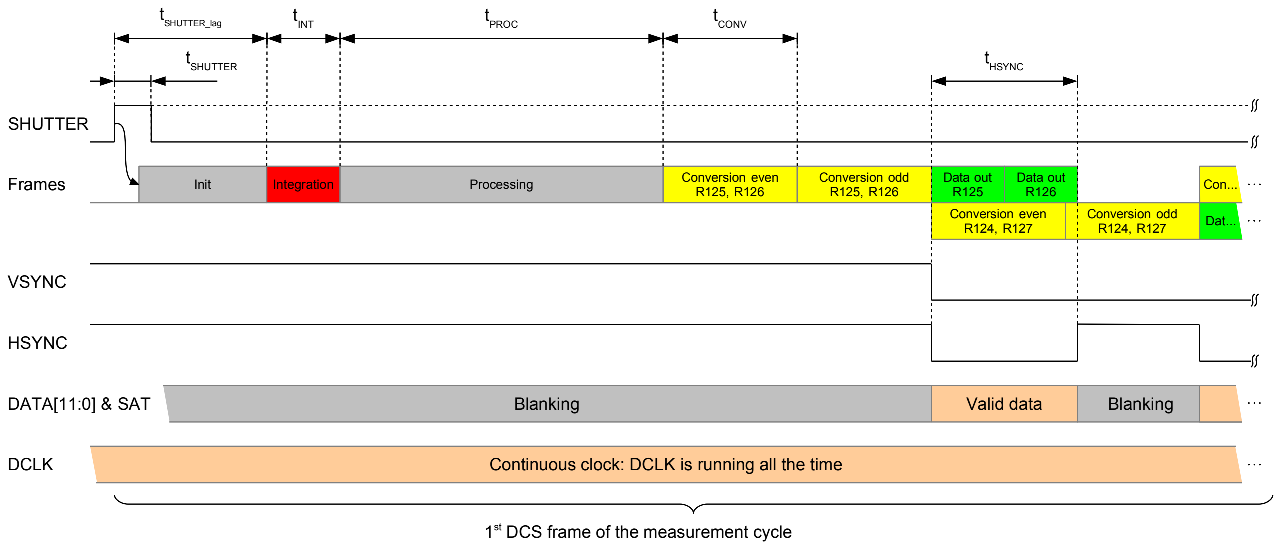



Figure 42: Frame timing: Start 1st DCS frame (DCLK: 40MHz)

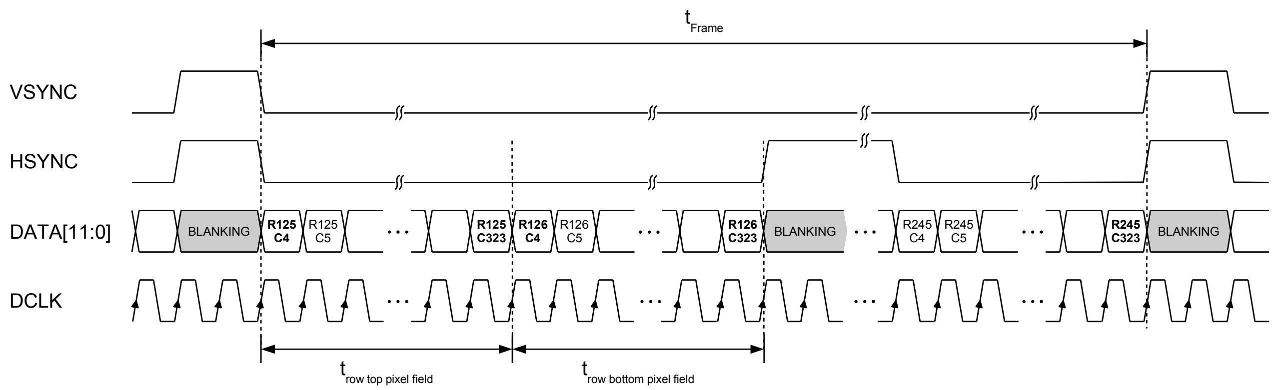



Figure 43: Data readout timing

### 5.8.2. Single or continuous measurement control

#### Single measurement control

The selected measurement mode (4x DCS, 2x DCS, black & white, …) defines, how many frames the chip performs by the stimulation of one SHUTTER pulse for a measurement cycle. This pulse can be applied either by the HW SHUTTER pin or by SW control of the shutter_en bit. Whereas the SW controlled SHUTTER is auto-cleared after propagation, the HW Shutter needs a minimum hold time of 250ns and must be set back manually latest before the HSYNC_A1 signal of the last row pair of the last DCS frame (last HSYNC_A1 of the last frame).

During such a measurement cycle, the next frame acquisition starts immediately after the last data readout on the TCMI interface until all frames are performed.

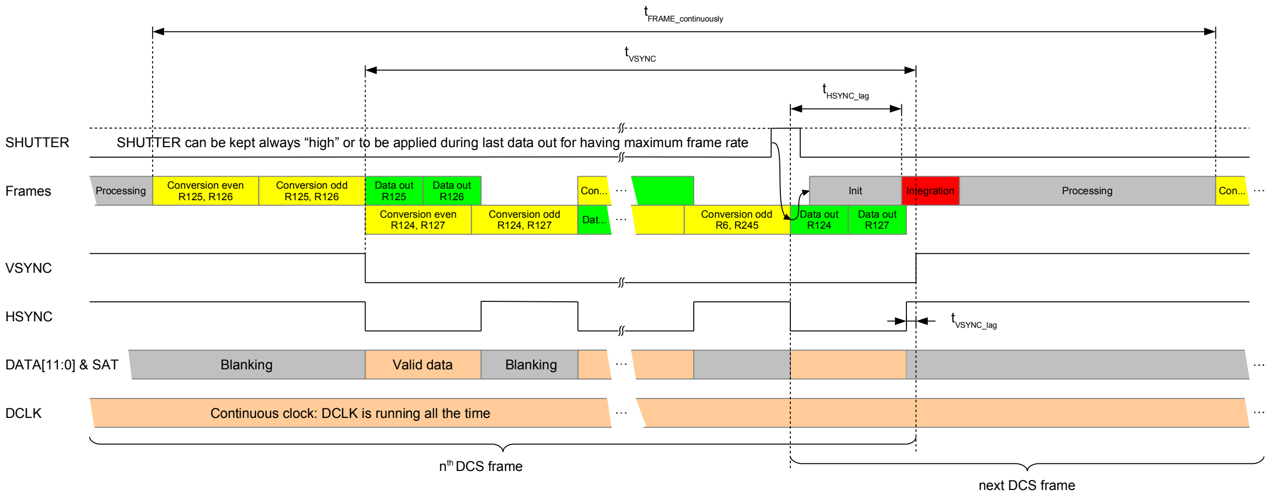



Figure 44: Frame timing: Inter frame timing, end of frame and start next frame (DCLK: 40MHz)

#### Continuous measurement control (auto-run)

As long as in the Shutter_Control register the multi_frame_en = 1 is set or the HW SHUTTER is applied during the readout of the last row pair of the last frame, the epc660 runs in a non-stop measurement mode. The chip starts immediately next measurement cycle if the actual one is terminated (Figure 44). If the trigger arrives before the readout of the last row pair of the last frame, then it is ignored.

### 5.8.3. Frame rates and data-out performance

**Default QVGA frame and 3D TOF 4x DCS distance measurement**

Frame rate:

The epc660 can perform a maximum of 262 fps (frames per second) in any mode with its full 320x240 pixel resolution (1µs integration time, 80MHz mod_clk, 40MHz DCLK, DLL off, 4x DCS, continuous measurement control).

For 3D TOF, each frame is referred as a DCS frame. Either 4x (with π-delay matching) or 2x (without π-delay matching) DCS frames must be acquired for one distance calculation. Therefore, the resultant distance measurement rate turns out to be 65.5 fps or 131 fps respectively.

For the black & white mode the maximum frame rate applies: 262 fps.

The TCMI transfers during HSYNC a burst of 2 rows at the time, which gives 2row x 320pixel x 13bits/pixel = 8'320 bits or 640 samples (pixels) in every burst transfer. This gives a total burst duration (HSYNC) of 32, 16 and 8µs at $f_{DCLK}$ = 20, 40 and 80MHz respectively.

The data-out latency is defined as the time from the SHUTTER = 1 (or shutter_en = 1 via $I^2C$) to the first sample data coming out of the TCMI interface, refer to Figure 42. This is <69µs (excluding integration time). For 1 µs integration time, data-out latency is 69.8µs. Latency is fixed for all binning and resolution reduction modes.

The time of one full 320x240 pixel DCS frame is defined as the time from the SHUTTER = 1 (or shutter_en = 1 via $I^2C$) to the last sample data coming out of the TCMI interface. For 1 µs integration time, this frame time is 3'807µs/frame (262fps). Note, frame time gets shorter (frame rate gets faster) linearly with binning and resolution reduction modes for very short integration times.

| Symbol | Parameter | Min. | | Max. | Units |
|---|---|---|---|---|---|
| $t_{DCLK}$ | TCMI readout clock e.g. $f_{DCLK}$ = 40MHz | | 25 | | ns |
| $t_{SHUTTER}$ | Hold time for the signal on pin SHUTTER | 250 | | | ns |
| $t_{SHUTTER\_lag}$ | Delay from the rising edge of SHUTTER signal to the 1st LED pulse | | 18 | | µs |
| $t_{INT}$ | Image acquisition (integration time) | | | | |
| $t_{PROC}$ | Delay from the last LED pulse until the 1st row conversion | | 38.75 | | µs |
| $t_{CONV}$ | Conversion time for a pair of half rows (even or odd) | | 15.625 | | µs |
| $t_{HSYNC}$ | Readout time for a pair of rows e.g. $f_{DCLK}$ = 40MHz | | 16 | | µs |
| $t_{HSYNC\_lag}$ | Delay from the begin of last readout until the 1st LED pulse of next DCS frame | | 17 | | µs |
| $t_{VSYNC\_lag}$ | Delay end of HSYNC to end of VSYNC at the end of each DCS frame | | 50 | | ns |
| $t_{VSYNC}$ | Data readout time for one DCS frame e.g. $f_{DCLK}$ = 40MHz<br>$t_{VSYNC} = (2x\ t_{CONV}\ x\ 119\ rows) + t_{HSYNC} + t_{VSYNC\_lag}$ | | 3'735 | | µs |
| | | | | | |
| | **Single measurement control mode:** | | | | |
| $t_{1st\_FRAME\_START}$ | Delay from rising edge of SHUTTER signal until start of data readout of 1st frame | | 3'824 | | µs |
| $t_{1st\_FRAME\_TOTAL}$ | Total time for reading one DCS or B&W frame from rising edge of SHUTTER signal until end of readout of 1st frame | | 3'890 | | µs |
| | | | | | |
| | **Continuous measurement control mode:** | | | | |
| $t_{FRAME\_continuously}$ | Total time for reading one DCS or B&W frame<br>$t_{FRAME\_continuously} = (2x\ t_{CONV}\ x\ 120\ rows) + t_{HSYNC\_lag} + t_{INT} + t_{PROC}$ | | 3'807 | | µs |
| $t_{4DCS\_continuously}$ | Total time for one 3D TOF distance measurement (4 DCS)<br>$t_{FRAME\_continuously} = ((2x\ t_{CONV}\ x\ 120\ rows) + t_{HSYNC\_lag} + t_{INT} + t_{PROC}\ )\ x\ 4\ DCS$ | | 15.23 | | ms |

Table 16: Timings for one DCS or B&W frames and for 3D TOF distance measurements (4x DCS)
(Reference: see Figure 42 and Figure 44, $f_{DCLK}$ = 40MHz, $t_{INT}$ = 1µs)

Note:

$t_{Init}$: In frame cycles having a DLL synchronization $t_{Init}$ = 37µs + $t_{SYNC}$. = 37µs + DLL pre-synchronization + DLL lock time e.g. $t_{Init}$ = 77µs.

| Action | Comments | Value | Units |
|---|---|---|---|
| DCS & B&W frame rate | 1 DCS or B&W frame | 262 | fps |
| Distance measurement frame rate | 4 DCS measurement | 65.5 | mps |

Table 17: Frame rates based on the Table 16 timing
(Continuous measurement control mode)

Memory space estimation:

The data-out operation transfers effectively 320x240pixel x 13bits/pixel (12 bit DATA + 1 bit SAT) = 998'400 bits per frame (~1Mbpf) with each frame via the TCMI interface.

In a typical application with a CPU having a high-speed parallel interface (GPIO, CPI, CMI, etc.), each 13 bit sample can be packed into a 2 byte word, then stored in the application's frame buffer (internal SDRAM of the CPU or external SDRAM bridged via CPU). In that case, a burst (2 rows) makes 1'280 bytes. The frame acquisition including all rows/columns requires a storage space maximum of 153'600 bytes or 150kbyte per frame (1 kbyte = 1'024 bytes). For the 3D TOF distance measurements with 4 DCS, the application frame buffer must be large enough to accommodate at least 4 DCS frames plus (may be) another 1 DCS frame size to store back the calculated distance data. For a smooth SW programming, doubling this space is highly recommended as a bare minimum for the application frame buffer size. This corresponds to a total of 2 x (4 + 1) x 150 kbytes = 1'500kbytes (~1.5 Mbytes) memory space. In practice, this can be realized with a single 2 Mbyte (or 2x 1Mbyte) SDRAM. Depending on the complexity of the application SW, this size must be increased.

Balance between frame rate, resolution, measurement dynamic and memory space can be adjusted by combining binning, resolution reduction and ROI modes together.

There is a comparison in Table 18 below to give an overview on the performance figures (conditions: 1µs integration time, 1 DLL measurement every 100 frames) in different resolution reduction modes with ROI set to QVGA (default: 320 x 240 pixel).

| Ref. | Imager settings (Input) | | | Imager output | | |
|---|---|---|---|---|---|---|
| Figure | MGX mode [1] | Binning [2] hor., ver., both | Row reduction y-axis: 2, 4, 8 | Resolution x-y [imager pixel] | Frame rate [3] [fps] | Frame size [4] [kbytes] |
| 35 | single | no | 1 | 320 x 240 | 262 | 150 |
| 35 | single | no | 2 | 320 x 120 | 524 | 75 |
| 35 | single | no | 4 | 320 x 60 | 1'048 | 37.5 |
| 35 | single | no | 8 | 320 x 30 | 2'096 | 18.75 |
| 36 | single | horizontal | 1 | 160 x 240 | 524 | 75 |
| 36 | single | horizontal | 2 | 160 x 120 | 1'048 | 37.5 |
| 36 | single | horizontal | 4 | 160 x 60 | 2'096 | 18.75 |
| 36 | single | horizontal | 8 | 160 x 30 | 4'192 | 9.375 |
| 37 | single | vertical | 2 | 320 x 120 | 524 | 75 |
| 37 | single | vertical | 4 | 320 x 60 | 1'048 | 37.5 |
| 37 | single | vertical | 8 | 320 x 30 | 2'096 | 18.75 |
| 38 | single | both | 2 | 160 x 120 | 1'048 | 37.5 |
| 38 | single | both | 4 | 160 x 60 | 2'096 | 18.75 |
| 38 | single | both | 8 | 160 x 30 | 4'192 | 9.375 |
| 39 | dual | no | 1 | 2 x 320 x 120 | 262 | 150 |
| 39 | dual | no | 2 | 2 x 320 x 60 | 524 | 75 |
| 39 | dual | no | 4 | 2 x 320 x 30 | 1'048 | 37.5 |

Table 18: Frame rate and resolution versus different modes of operation for default QVGA mode
Default ROI setting: 320 x 120 pixel: top-left (4, 6) and bottom-right (323, 125)

Notes:

[1] When dual MGX mode is turned on, resolution, frame rate and frame size are identical for both MGX modes (same integration time, motion blur reduction or with different integration times, high dynamic range).

[2] Dual MGX mode and binning together is not possible.

[3] Frame rate is the DCS frame rate.

[4] Frame size is based on 12 bit DATA[11:0] + 1 bit SAT flag per sample packed in 2 Bytes to store in the application frame buffer.

**Half QQVGA frame (160x60 pixel)**

The second example shows the ROI set symmetrically to 2 x 160 x 30 (Half-QQVGA) in the middle of the pixel field.

The frame time scales linearly with the reduced number of rows readout (see Table 19).

The TCMI data-out time scales linearly with the reduced number of columns set in the ROI.

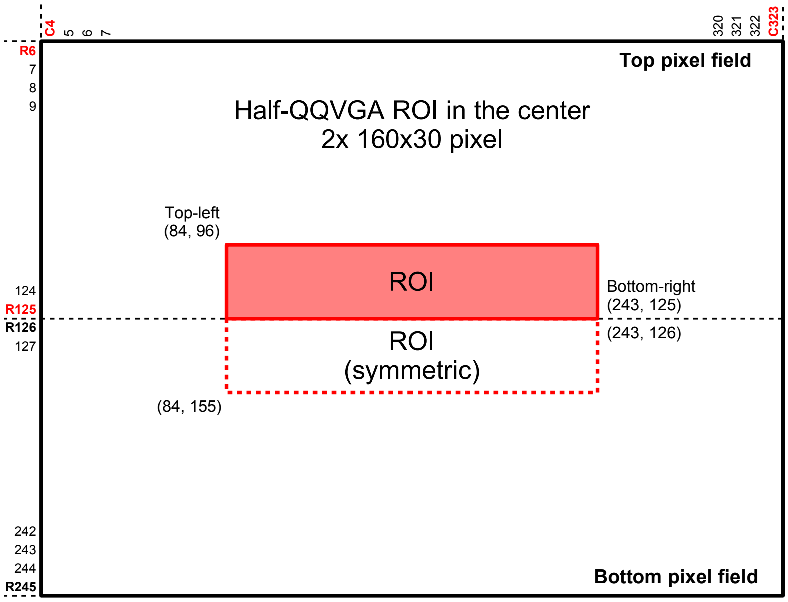



Figure 45: ROI for Half-QQVGA: 2 x 160 x 30 pixel

| Ref. | Imager settings (Input) | | | Imager output | | |
|---|---|---|---|---|---|---|
| Figure | MGX mode | Binning [1] hor., ver., both | Row reduction y-axis: 2, 4, 8 | Resolution x-y [imager pixel] | Frame rate [2] [fps] | Frame size [3] [kbytes] |
| 35 | single | no | 1 | 160 x 60 | 1'048 | 18.75 |
| 35 | single | no | 2 | 160 x 30 | 2'096 | 9.375 |
| 36 | single | horizontal | 1 | 80 x 60 | 2'096 | 9.375 |
| 36 | single | horizontal | 2 | 80 x 30 | 4'192 | 4.6875 |
| 37 | single | vertical | 2 | 160 x 30 | 2'096 | 9.375 |
| 38 | single | both | 2 | 80 x 30 | 4'192 | 4.6875 |
| 39 | dual | no | 1 | 2 x 160 x 30 | 1'048 | 18.75 |

Table 19: Frame rate and resolution versus different modes of operation for default Half-QQVGA mode
ROI setting: 160 x 30 pixel: top-left (84, 96) and bottom-right (243, 125)

Notes:

[1] Dual MGX mode and binning together is not possible.

[2] Frame rate is the DCS frame rate.

[3] Frame size is based on 12 bit DATA[11:0] + 1 bit SAT flag per sample packed in 2 Bytes to store in the application frame buffer.

Note: For the emulation of the epc635 refer to the application section: chapter 7.6.4., epc635 emulation.

### 5.8.4. Example applications of CPU architectures for high-speed frame data transfer

Several possible application CPU, DSP, FPGA configurations can be used with the epc660. Here are two architectures proposed as examples, to give the application developer a high-level overview.

The first one is a completely symmetric, non-blocking application frame buffer CPU architecture with a high-throughput imager to application frame buffer data transfer concept. While the frame is continuously acquired, the current frame data is streamed from the epc660 to the application frame buffer A. At the same time,the CPU can access and calculate the previously acquired data from the application frame buffer B. The next frame, the memory channels are dynamically swapped between epc660 and the CPU by re-configuring the DMA channels. As there are two physically identical, separated memory channels, both transfers can flawlessly happen at the same time (see Figure 46).

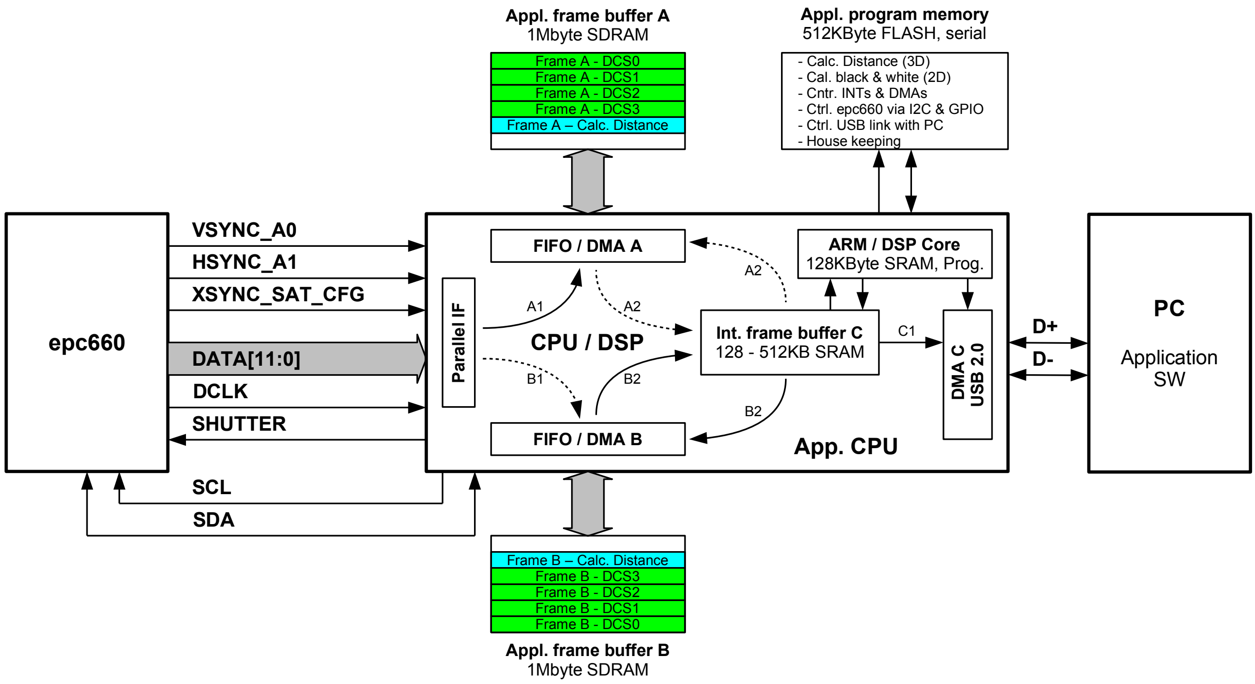



Figure 46: Symmetric, non-blocking application frame buffer CPU architecture

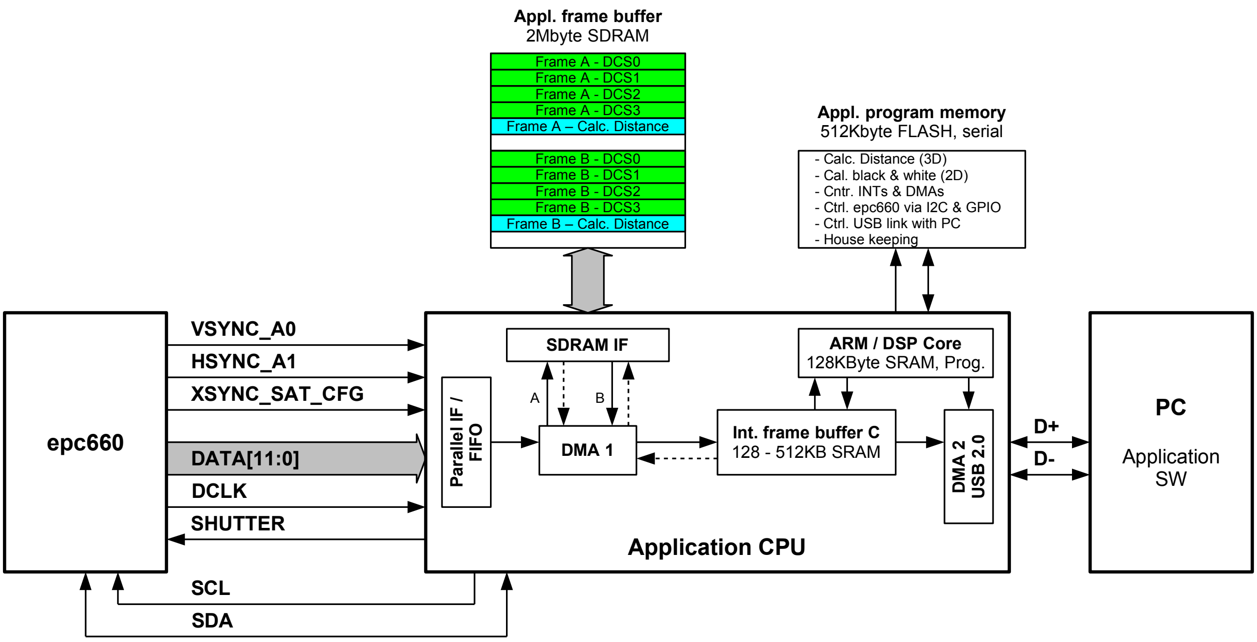



Figure 47: Time-shared application frame buffer CPU architecture

The second example is a time-shared application frame buffer CPU architecture. The epc660 must have a higher priority over CPU (see Figure 47) when accessing the application frame buffer SDRAM. The memory is blocked to the CPU as long as epc660 is transferring the burst. At the end of the burst, the memory channel is released by the DMA controller and the CPU gets access granted. In the time-shared bus architecture, the application developer must carefully tune the bus access ratio between the epc660 and the CPU. This can be adjusted by setting a different TCMI DCLK frequency (20, 40 or 80MHz) on the epc660 side. The higher the TCMI bus frequency, the faster the burst is transmitted (min. burst duration for full pixel rows: 8µs at $f_{DCLK}$= 80MHz).

Another option is using a CPU having a much wider and faster SDRAM interface. Example:

Assume using a CPU which has a 128 bit wide SDRAM bus running at 160MHz clock. The TCMI DCLK is set at 40MHz. This gives 4x performance increase between FIFO → application SDRAM due to the TCMI to SDRAM bus clock ratio. Another 8x increase is due to 16-bits packed samples in the FIFO and transferred over a 128 bit wide bus to the SDRAM. It results in a total of 32x faster transfers between FIFO → application SDRAM, application SDRAM → internal SRAM of the CPU and vice-versa. This gives a lot of bandwidth to the CPU to transfer data back and forth between the application SDRAM and the internal SRAM between computations.

### 5.9. Power consumption levels

The epc660 has mainly 7 power states/levels during the different operation phases. RESET state is the lowest, INTEGRATE is the highest average power consumption level.

| Power state | Power [mW] | Operation description |
|---|---|---|
| RESET | 54 | All supplies are ON, $\overline{\text{RESET}}$ = 0,<br>Oscillator is ON, PLL and all system system clocks are OFF |
| READY | 81 | $\overline{\text{RESET}}$ = 1, PLL and all system clocks ON, waiting for SHUTTER |
| INTEGRATE | 1'300 | SHUTTER = 1 or shutter_en= 1 via $I^2C$, integrating |
| CONVERSION | 580 | Integration finished, conversion of rows |
| CONVERSION + DATAOUT | 555 | Transmit row data via TCMI while converting next row |
| DATAOUT | 110 | Transmit last row data via TCMI |

Table 20: Typical average power consumption levels at different operating states (integration time < 5ms)

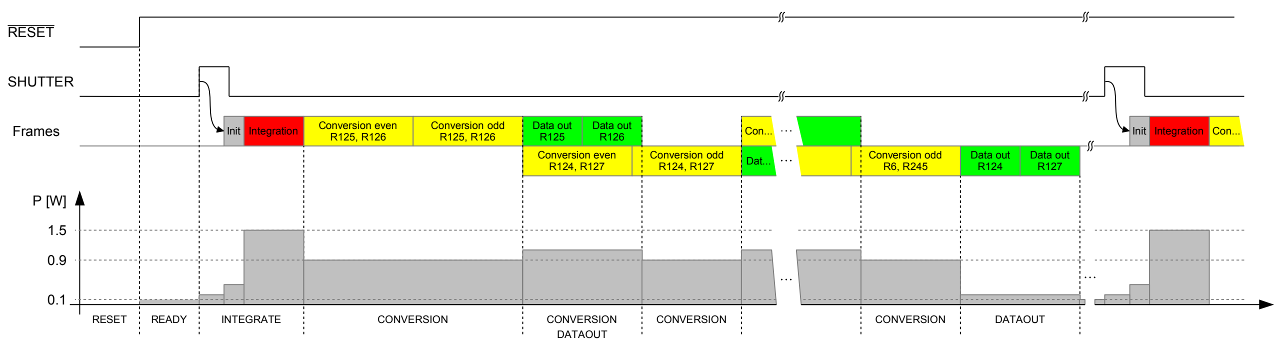



Figure 48: Power consumption levels and operating states

# 6. Measurement Modes

### 6.1. Distance Measurement Modes (3D TOF)

The distance measurement modes use the on chip LED driver and the external IR-LED/LD to provide modulated light on the target. Modulation control signals to the LED driver are provided by the programmable modulator (see chapter 6.1.1.). Two modulation sequences are supported: Sine mode and PN mode.

Sine mode can be used for single camera environments. PN mode can be used for multi-camera environments.

#### 6.1.1. Modulator

The modulator (see Figure 27) generates all signals to modulate the external IR-LED/LD via the LED driver and simultaneously all demodulation signals to the pixel field for performing a measurement. Sine, PN and black & white mode with all the variants are generated here.

The mode selection takes place by the modulation table registers (refer to Table 41). More details to these modes are listed in following the chapters and in Table 26. The registers can be updated via I$^2$C bus between the frame acquisitions. The application must take care that the last frame's integration phase is completed before modifying these registers on the fly. This time can be detected by the application by waiting for the falling-edge of VSYNC or the first falling-edge of HSYNC signal after SHUTTER = 1 was applied. This allows to run continuously at the maximum frame rate. For a full-frame readout, the margin is a 3.6ms to alter these registers via I$^2$C on the fly. This margin scales linearly with the number of row readouts.

#### 6.1.2. Sine mode (Sinusoidal modulation)

The epc660's default modulation mode is the sinusoidal modulation. After reset, all internal register values are default to operate the chip: at 4MHz XTAL/external clock input, multiplied up to 80MHz at the PLL output, clocks the modulator with 80MHz internal mod_clk, modulates LED/LD with 20MHz and acquires 4 successive DCS frames (0 ... 3) using 25.6µs integration time.

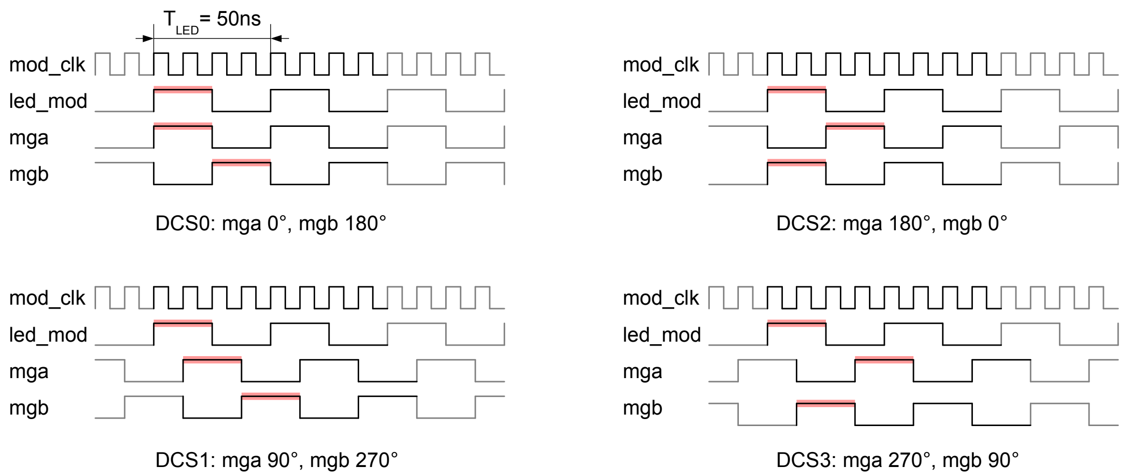



Figure 49: Sine mode 4x DCS modulation/demodulation waveforms (mod_clk = 80MHz)

With the application of the shutter trigger pulse (HW SHUTTER = 1 or SW shutter_en = 1 via I$^2$C), the chip performs 4 successive DCS (Difference Correlation Sample) frame acquisitions. Each one of the 4 DCS frame types has a different phase relation between modulation (led_mod) and demodulation (mga, mgb) signals which makes phase-to-distance calculation possible. In case of DCS0, led_mod is phase-shifted by 0º and 180º with respect to mga and mgb, respectively. In case of DCS1, led_mod is phase-shifted by 90º and 270º. For DCS2, the phase shifts are 180º and 0º and for DCS4, the phase shifts are 270º and 90º (see Figure 49). Note that for DCS2 and DCS3, the demodulation signals mga and mgb are simply swapped with respect to DCS0 and DCS1, respectively.

By programming dcs_sel = 01 (see MOD_Control register in Table 42, Modulator/demodulator registers), shutter trigger initiates 2 successive DCS frame acquisitions (see Figure 50).

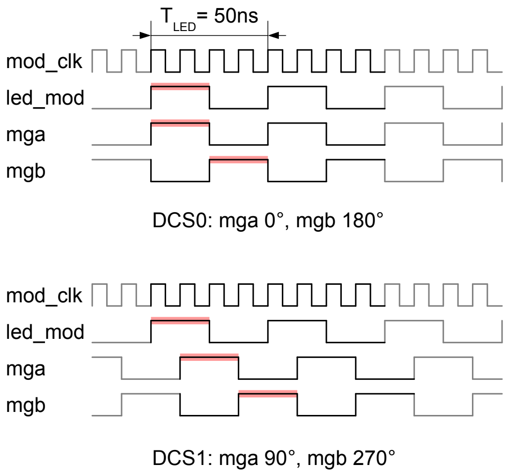


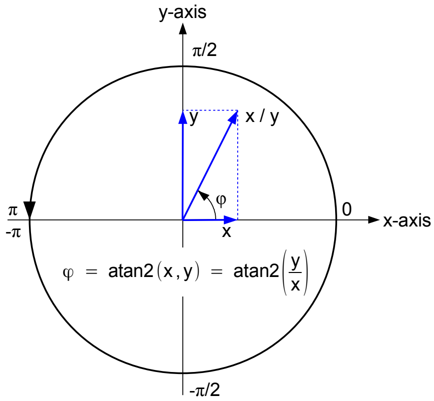

Figure 50: Sine mode 2x DCS modulation/demodulation waveforms
(mod_clk = 80MHz)

The first sample pair (DCS0 and DCS1) is enough to calculate 3D TOF distance for that pixel. The second pair (DCS2 and DCS3) can be used for calculating the π-delay matching algorithm to compensate for fixed pattern noise and device mismatches during pixel readout operation. The application SW must run the same algorithm on every pixel to compute the 3D TOF distance for the entire pixel field.

**Sine mode: Distance calculation algorithm**

The use of the atan definition of the Cartesian coordinate system (atan2) guarantees a continuous distance calculation algorithm in the range of phases between -π … +π. In our case, we use the range from 0°… 360° which corresponds to the distance from 0m up to the unambiguity distance.

Definition atan2 for the range -π … π




$$\operatorname{atan2}(x,y) := \begin{cases} \operatorname{atan}(y/x) & \text{for } x > 0 \\ \operatorname{atan}(y/x) + \pi & \text{for } x < 0,\ y \geq 0 \\ \operatorname{atan}(y/x) - \pi & \text{for } x < 0,\ y < 0 \\ +\pi/2 & \text{for } x = 0,\ y > 0 \\ -\pi/2 & \text{for } x = 0,\ y < 0 \\ 0 & \text{for } x = 0,\ y = 0 \end{cases}$$

Note: Variations in notation

$\varphi = \operatorname{atan2}(x, y)$ or $\operatorname{atan2}(y, x)$

The arguments are reversed depending on implementation.

Figure 51: Continuous atan representation for the range -π … +π

Let the respective 12 bit samples measured for a single pixel from 4 DCS (0, 1, 2, 3) frames be DCS0, DCS1, DCS2 and DCS3.

Thereof, the sampling of 4x DCS samples corresponds to the following sinusoidal signal representation for the epc660 chip:

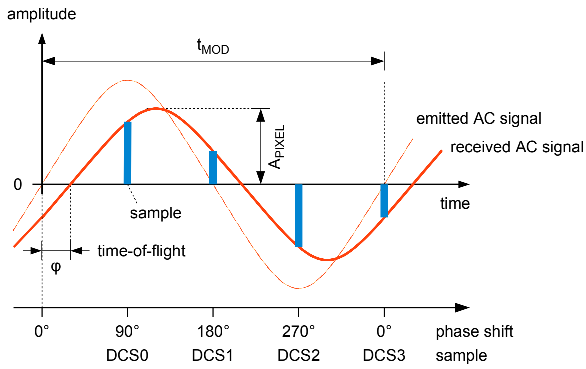



Figure 52: Sampling of the received waveform

The following terms are used for the formulas:

| | |
|---|---|
| $D_{TOF\ sine}$ | Distance in meters [m] |
| c | Speed of light e.g. 300'000'000m/s |
| $f_{LED}$ | LED/LD modulation frequency e.g. 20MHz |
| DCS0 - DCS3 | Sampling amplitude [LSB] |
| φ | Phase shift caused by the time-of-flight [rad] |
| $D_{OFFSET}$ | Offset compensation [m] |

In the real case of 4x DCS samples used with π-delay matching algorithm, the time of flight calculation will be:

[1] $$t_{TOF\ sine}[sec] = \frac{1}{2\pi f_{LED}} \cdot [\pi + atan2(DCS2 - DCS0, DCS3 - DCS1)] + t_{OFFSET}$$

then, the distance calculation will be (c is the speed of light):

[2] $$D_{TOF\ sine}[m] = \frac{c}{2} \cdot \frac{1}{2\pi f_{LED}} \cdot [\pi + atan2(DCS2 - DCS0, DCS3 - DCS1)] + D_{OFFSET}$$

Based on the ideal case of 2x DCS samples (without any offsets), the time of flight may be computed as follows:

[3] $$t_{TOF\ sine}[sec] = \frac{1}{2\pi f_{LED}} \cdot [\pi + atan2(-DCS0, -DCS1)]$$

then, the distance calculation is:

[4] $$D_{TOF\ sine}[m] = \frac{c}{2} \cdot \frac{1}{2\pi f_{LED}} \cdot [\pi + atan2(-DCS0, -DCS1)]$$

**Sine mode: Quality of the measurement result**

The epc660 provides information on the quality and the validity of the received optical signal. This reflects the confidence level of the measurement result. The better the received signal, the better and more precise the distance measurement will be.

Each distance measurement of every pixel has its own validity and quality.

The primary quality indicator for the measured distance data is the peak-to-peak amplitude value of the received modulated light $A_{TOF\ sine\ PP}$.

After each measurement, this needs to be calculated from the DCSx values delivered by the chip. This amplitude value is the feedback parameter that is used to set the integration time for the next measurement.

[5] $$A_{TOF sine PP} = \sqrt{\frac{(DCS0 - DCS2)^2}{4} + \frac{(DCS3 - DCS1)^2}{4}}$$

| Amplitude $A_{TOF\ sine\ PP}$ | Classification | Action |
|---|---|---|
| < 25 LSB | Weak illumination | Too less signal for an accurate measurement:<br>Increase integration time for the next measurement |
| 25 … 100 LSB | Enough signal for a useful measurement | Distance noise approx. 4 times higher than the optimum:<br>No action necessary. See note below |
| 100 … 1'200 LSB | Good signal strength | No action necessary. See note below |
| > 1'200 LSB | Overexposed | Decrease integration time for the next measurement.<br>See note below |

Table 21: Signal amplitude versus classification

Note:
Generally, the higher the received signal, the better and more precise the distance measurement will be. However, it is good practice to control the integration time such that an amplitude value between 100 ... 200 LSB is achieved. Higher values will only slow down the acquisition rate due to longer integration times, but are not significantly improving signal to noise ratio.

The quality indicator for the distance noise is the ratio of ambient-light $E_{BW}$ to the peak-to-peak value of modulated light $E_{TOF\ PP}$ (AMR). This value may be calculated and used additionally to the above amplitude value if the respective application is subject to intense ambient light.

The peak-to-peak irradiance $E_{TOF\ sine\ PP}$ of the modulated signal at the surface of a pixel can be calculate out of the AC sensitivity $S_{TOF\ sine}$, the used integration time $t_{INT\text{-}TOF}$, the reference integration time $t_{INT\text{-}REF\text{-}TOF}$ and the peak-to-peak amplitude $A_{TOF\ sine\ PP}$ of the received modulated signal the following way:

[6] $$E_{TOF\ sine\ PP} = S_{TOF\ sine} \cdot \frac{t_{INT\text{-}REF\text{-}TOF}}{t_{INT\text{-}TOF}} \cdot A_{TOF\ sine\ PP} \quad \text{e.g.} \quad E_{TOF\ sine\ PP} = 155\frac{nW/cm^2}{LSB} \cdot \frac{103\mu s}{205\mu s} \cdot 1'000\ LSB = 78\mu W/cm^2$$

The formula to calculate the quality indicator “Ratio of ambient-light / modulated light” (AMR) is

[7] $$AMR[dB] = 20 \cdot \log\left(\frac{E_{BW}}{E_{TOFPP}}\right) \qquad \text{e.g.} \quad AMR[dB] = 20 \cdot \log\left(\frac{3.96\,mW/cm^2}{78\,\mu W/cm^2}\right) = 34dB$$

Refer for $E_{BW}$ to chapter 6.2. Black & white mode.

This ratio is one of the influencing factors regarding the distance noise.

| AMR value | Classification | Action |
|---|---|---|
| < 60 dB | excellent | No action necessary. |
| < 70 dB | sufficient | Is a lower noise level needed, do the next measurement with a longer integration time or with an increased illumination power. |
| > 70 dB | weak | Do the next measurement with a longer integration time or with an increased illumination power. |

Table 22: Classification ratio ambient-light to modulated light (AMR) versus distance noise

There are also validity indicators delivered by the chip after a measurement. These will help to detect saturated or not illuminated pixels as a result of too much/less illumination or too long/short integration time.

| Validity indicator (per pixel) | Classification | Action |
|---|---|---|
| Saturation flag:<br>= set | Pixel saturation:<br>The pixel receives too much light (too much modulated light-signal, too reflective object or too much ambient-light)<br>Refer to chapter 5.6.4. Pixel saturation detection and pin XSYNC_SAT_CFG. | Dump data and repeat the measurement with a decreased integration time or measure the ambient-light. |
| one or more of the DCSx values:<br>\| DCSx \| > 1'200 LSB | DCSx value over maximum signal limit:<br>The object is too reflective( It is too close to the sensor, there is too bright illumination or too much modulation signal is emitted) | Dump data and repeat the measurement with a decreased integration time. |
| all DCSx values:<br>\| DCSx \| ≤ 25 LSB | DCSx values below noise level limit:<br>Absence of an object in the operating range.<br>The object has too little remission (reflectivity): It is too far away, the illumination is too dim or too little modulation signal is emitted) | Dump data and repeat the measurement with an increased integration time. |

Table 23: Validity indicator versus classification

Table 23 shows a quality decision matrix as a summary of the validity and quality parameters for the distance measurement.

| Step | Sensor status | Pixel saturation<br><br>XSYNC_ SAT | Maximum signal<br><br>min. one \| DCSx \| | Noise level<br><br>all \| DCSx \| | Modulated light amplitude<br><br>$A_{TOF\,PP}$ | Ratio ambient to modulated light<br><br>AMR | Action |
|---|---|---|---|---|---|---|---|
| 1 | Saturation or bright object within scene | flag set | | | | | Repeat measurement with decreased integration time and/or illumination |
| 2 | Saturation or bright object within scene | no | > 1'200 LSB | | | | Repeat measurement with decreased integration time and/or illumination |
| 4 | No object detected | no | < 1'200 LSB | < 25 LSB | | | Repeat measurement with increased integration time or illumination |
| 3 | Overexposure or bright object within scene | no | < 1'200 LSB | > 25 LSB | > 1'200 LSB | | Repeat measurement with decreased integration time or illumination |
| 4 | No object detected | no | < 1'200 LSB | > 25 LSB | < 25 LSB | | Repeat measurement with increased integration time or illumination |
| 5 | Too much ambient-light | no | < 1'200 LSB | > 25 LSB | 25 LSB … 1'200 LSB | > 60 db (or > 70 dB) | Repeat measurement with increased integration time or illumination |
| 6 | Object detected | no | < 1'200 LSB | > 25 LSB | Good: 100 … 1'200 LSB<br>Enough: 25 … 100 LSB | < 60 db (or < 70 dB) | No action necessary |

Table 24: Quality decision matrix

### 6.1.3. PN mode (Pseudo-random Noise modulation)

The epc660's PN mode is initialized via the MOD_Control register, setting the mod_sel = 01. This activates the PN mode to create pseudo-random 0s and 1s sequences to the modulator to drive the led_mod modulation signal and mga, mgb demodulation signals.

In a multi-camera application, if all cameras are triggered at the same time (i.e. same signal drives all SHUTTER inputs on the PCB), each one must be initialized with a different type/length of PN sequence.

Similar to sine mode, the modulator generates 4 different DCS frames using the pseudo-random sequences (see Figure 53).

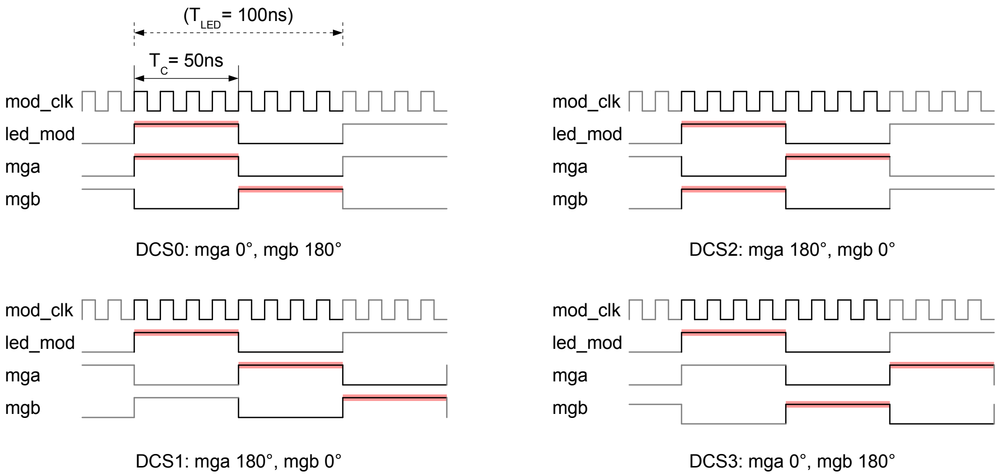



Figure 53: PN mode 4x DCS modulation/demodulation waveforms
(mod_clk = 80MHz)

By programming dcs_sel = 01 (see MOD_Control register in Table 42, Modulator/demodulator registers), shutter trigger initiates 2 successive DCS frame acquisitions (see Figure 54).

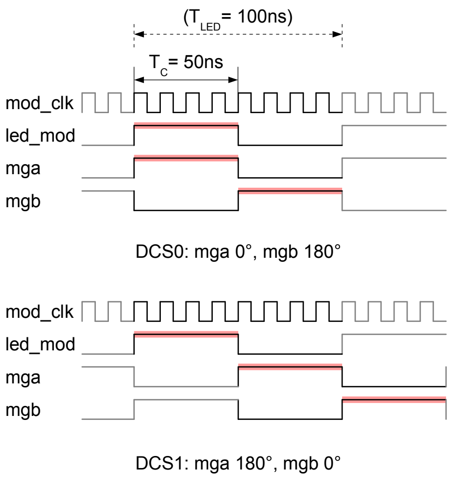



Figure 54: PN mode 2x DCS modulation/demodulation waveforms
(mod_clk = 80MHz)

**PN mode: Distance calculation algorithm:**

**Functionality**
Theoretically, the distance information can be obtained through the measurement of two orthogonal correlation values $C_0$ and $C_1$. The acquisition timing and processing of the correlation samples depend on the type of modulation that is implemented. If PN modulation is used, the two samples are acquired at time t and t+$T_c$. Tc is the "chip period" (1 bit / fraction period) of the PN modulating sequence. The minimum integration time or the step for larger integration times is equal to the PN sequence length.

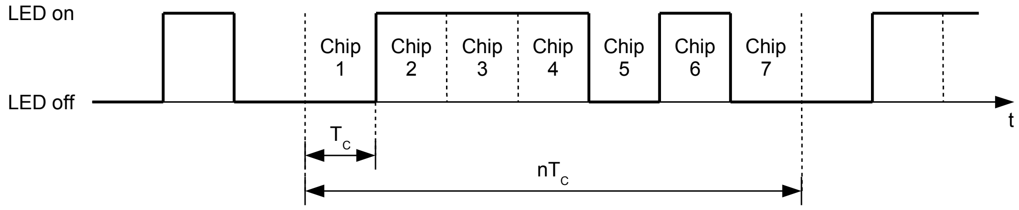



Figure 55: Example of PN modulation sequence
Maximal PN sequence length of n = 7 chips,
$T_C$: Chip period,
$nT_C$: Duration of a PN sequence = basic integration time

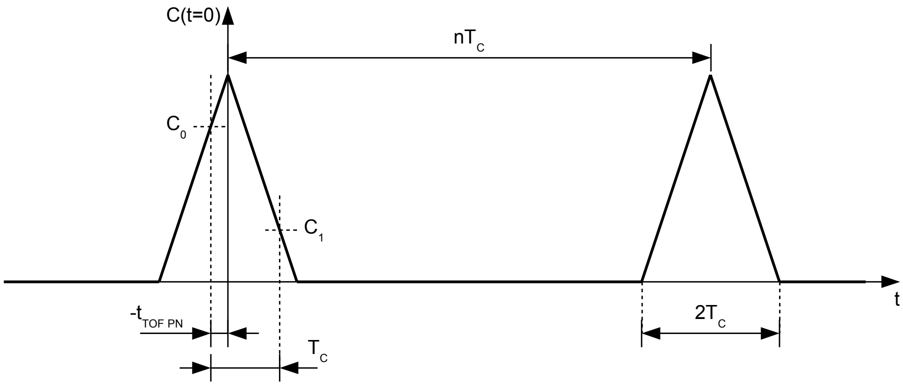



Figure 56: Auto-correlation function between the optical signal and the reference pixel signal

In order to compensate for mismatches, a swap of the two demodulation gates for each correlation sample and an averaging is operated so that 4 correlation samples are effectively used for a single distance measurement evaluation.

Let the respective samples measured for a single pixel from 4 DCS frames (0 … 3) be DCS0, DCS1, DCS2 and DCS3. According to the above definition, the correlation samples are named as:

■ DCS0: $C_0$ correlation sample at given time t.

■ DCS1: $C_1$ correlation sample at given time t+$T_C$ for PN modulation.

■ DCS2: $C_2$ correlation sample at given time t with swapped signals between the pixel demodulation gates.

■ DCS3: $C_3$ correlation sample at given time t+$T_C$ for PN modulation with swapped signals between the pixel demodulation gates.

In the real case of 4x DCS samples used with π-delay matching algorithm, the time of flight may be computed as follows:

[8] $$t_{TOF\ PN}[sec] = \frac{|DCS1 - DCS3|}{|DCS0 - DCS2| + |DCS1 - DCS3|} \cdot T_C + t_{OFFSET}$$

In the case of 4 correlation samples (where DCS2 is the equivalent of DCS0 obtained by gates swap and DCS3 is the correspondent of DCS1), then the distance calculation results as follows:

[9] $$D_{TOF\ PN}[m] = \frac{|DCS1 - DCS3|}{|DCS0 - DCS2| + |DCS1 - DCS3|} \cdot D_{TOF\ max} + D_{OFFSET}$$

with maximum detectable distance range $D_{TOF\ max}$ is given as

[10] $$D_{TOF\ max}[m] = \frac{c \cdot T_C}{2}$$ with c = 300'000'000 m/s (speed of light) and $T_C$.

and the unambiguity distance

[11] $$D_{TOF\ unamb}[m] = n_p \cdot \frac{c \cdot T_C}{2}$$ with the PN sequence bit length $n_p$.

Based on the ideal case of 2x DCS samples, the time of flight may be computed as follows:

[12] $$t_{TOF\ PN}[sec] = \frac{|DCS1|}{|DCS0| + |DCS1|} \cdot T_c$$

then, the distance calculation will be:

[13] $$D_{TOF\ PN}[m] = \frac{|DCS1|}{|DCS0| + |DCS1|} \cdot D_{max}$$

**PN mode: Quality of the measurement result**

The same rules for the quality and validity of the data apply as given in the section "Sine mode: Quality of the measurement result".

The formula for the peal-to-peak amplitude $A_{TOF\ PN\ PP}$ in PN mode is

[14] $$A_{TOF\ PN\ PP} = \frac{|DSC0 - DCS2| + |DCS1 - DCS3|}{2}$$

The peak-to-peak irradiance $E_{TOF\ PN\ PP}$ of the modulated signal at the surface of a pixel can be calculate out of the AC sensitivity $S_{TOF\ PN}$, the used integration time $t_{INT\text{-}TOF}$, the reference integration time $t_{INT\text{-}REF\text{-}TOF}$ and the peak-to-peak amplitude $A_{TOF\ PN\ PP}$ of the received modulated signal the following way:

[15] $$E_{TOF\ PN\ PP} = S_{TOF\ PN} \cdot \frac{t_{INT\text{-}REF\text{-}TOF}}{t_{INT\text{-}TOF}} \cdot A_{TOF\ PN\ PP} \quad \text{e.g.} \quad E_{TOF\ sine\ PP} = 155\frac{nW/cm^2}{LSB} \cdot \frac{103\mu s}{205\mu s} \cdot 1'000\ LSB = 78\mu W/cm^2$$

**PN initialization**

The number of the cameras that can concurrently operate and the quality of the measurement depend on the PN sequence length. Generally, longer PN sequence brings better measurements with larger number of cameras. On the other hand, longer PN sequences prolong the response time (time until a valid distance signal is available) and the amplitude. The minimum integration time and the step for larger integration times is equal to the PN sequence length.

[16] $$t_{INT\ min} = n_p \cdot T_C \qquad \text{e.g.} \quad t_{INT\ min} = 255 \cdot 50ns = 12.75\mu s$$

The number of the LFSR stages ($m_p$) is set by lfsr_sel. It directly defines:

- length of the PN sequence ($n_p$)
- number of different PN sequences of the maximum length (m-sequences) that can be used ($M_p$)

The number of the primitive polynomials (m-sequences) depends on the number of the LFSR stages. In multi-camera environment, each camera must use a unique PN polynomial. The following table shows how many independent cameras can be used for a given LFSR. For each $m_p$ several primitive polynomials are given as example (section 7.10. describes how to generate more primitive polynomials).

| Number of LFSR stages ($m_p$) | PN sequence bit length ($n_p$) | Number of m-sequences for independent cameras ($M_p$) | Primitive polynomial (pn_poly) [1] | Integration time (Int_len_hi/lo) | Integration time needed for one PN sequence [µs] [2] |
|---|---|---|---|---|---|
| 8 | 255 | 16 | 0x11D , 0x12B, 0x12D, 0x14D, … 0x1F5 | 0x03FB | 12.75 |
| 9 | 511 | 48 | 0x211, 0x21B, 0x221, 0x22D, … 0x3FB | 0x07FB | 25.55 |
| 10 | 1'023 | 60 | 0x409, 0x41B, 0x427, 0x42D, … 0x7F9 | 0x0FFB | 51.15 |
| 11 | 2'047 | 176 | 0x805, 0x817, 0x82B, 0x82D, … 0xFE9 | 0x1FFB | 102.35 |
| 12 | 4'095 | 144 | 0x1053, 0x1069, 0x107B, 0x107D, … 0x1FC9 | 0x3FFB | 204.75 |
| 13 | 8'191 | 630 | 0x201B, 0x2027, 0x2035, 0x2053, … 0x3FFD | 0x7FFB | 409.55 |
| 14 | 16'383 | 756 | 0x402B, 0x4039, 0x4053, 0x405F, … 0x7FE7 | 0xFFFB | 819.15 |

Table 25: Properties of PN mode m-sequences

Notes:

[1] The remaining pn_poly register values which are not listed in the table can be calculated by the formula given in chapter 7.10., Calculation of PN mode polynomials (m-sequences). It is only the first 4 and the last computed numbers listed here as a quick reference for the application developer.

[2] mod_clk = 80MHz, $T_C$ = 50ns

**LFSR stage selection:**

The epc660 contains several LFSR implementations (each with different number of stages $m_p$). The default implementation is an 11-bit LFSR, set by lfsr_sel = 100 in the MOD_Control register.

**PN polynomial selection:**

After the number of stages of the LFSR is selected, the appropriate PN sequence length must be programmed. It should have the maximum length (m-sequence). PN polynomial size is selected through PN_POLY_hi/lo registers, The starting point of the PN sequence is initialized through PN_INIT_hi/lo registers (see Table 42, Modulator/demodulator registers). Since the PN modulation has good properties (i.e. high auto-correlation, low cross-correlation) only when the complete sequence is used, the integration length should be set accordingly (Int_len_hi/lo). It is allowed to use integration time during which multiple PN sequences are shifted.

### 6.2. Black & white mode

The black & white mode allows using the epc660 as a grayscale imager. This mode can be used either without LED/LD illumination for ambient-light measurements or with LED/LD for active illumination of the scenery.

The black & white measurement uses the regular DCS measurement but only with DCS0. It is performed in single-ended mode with readout of MGA. Refer to chapter 8.6.5, MT registers. Corresponding registers settings can be found in Table 26.

There will be a one frame acquisition with the 12 bit black & white values.

Due to the fact that distance measurement results can be influenced by ambient-light, the black & white measurement without illumination can thereof be used as an important quality and correction parameter for the distance measurement.

**Black & white: Quality of the measurement result**

The same rules apply for data quality and validity as given in section "Sine mode: Quality of the measurement result". The saturation flag status is invalid in this mode.

These black & white values can also be used for estimating quality information during sine/PN mode for the 3D TOF distance calculations.

The irradiance $E_{BW}$ of the grayscale signal at the surface of a pixel can be calculate out of the DC sensitivity $S_{BW}$, the used integration time $t_{INT\text{-}BW}$, the reference integration time $t_{INT\text{-}REF\text{-}BW}$ and the amplitude DCS0 of the grayscale signal the following way:

[17] $$E_{BW} = S_{BW} \cdot \frac{t_{INT\text{-}REF\text{-}BW}}{t_{INT\text{-}BW}} \cdot DCS0 \qquad \text{e.g.} \quad E_{BW} = 61.5\frac{nW/cm^2}{LSB} \cdot \frac{103\mu s}{1.6\mu s} \cdot 1'000\ LSB = 3.96\,mW/cm^2$$

# 7. Application information

As a help for the user to have an easier understanding of the chip, this chapter list a variety of typical application examples and their configurations for the epc660.

### 7.1. Example sequence from the start-up to frame acquisition

1. Apply all positive supplies, while keeping $\overline{RESET}$ = 0.
2. Wait until all positive supplies reach their rated levels.
3. Apply the $V_{BS}$ negative supply, while keeping $\overline{RESET}$ = 0.
4. Wait until $V_{BS}$ reached its rated level.
5. Optional: Set/check the external 10kOhm pull-up resistors on the strap pins (HSYNC_A1, VSYNC_A0).
6. Release $\overline{RESET}$ = 1.
7. Wait until the start-up/reset sequence is over ($t_{Strap\_scan}$ + $t_{EEPROM\_to\_CFG\_copy}$).
8. Optional: Program the TCMI interface signal polarities with respect to the application CPU interface requirements via I²C interface.
9. Do the counterpart for the parallel data interface settings on the application CPU.
10. Optional: Set LED driver and DLL properties and polarities with respect to external LED/LD circuit on the PCB via I²C. Set DLL measurement rate via I²C (DLL_measurement_rate_hi/lo register).
11. Select the measurement mode out of the Sine, PN or black & white modes (Default: Sine mode, 4 DCS).
12. Optional: If PN mode is selected, set also the number of LFSR stages (MOD_Control register, lfsr_sel bits), the PN polynomial (PN_POLY_hi/lo, PN_INIT_hi/lo registers) and the number for the distance offset (Dist_offset register, 0-63 steps of $t_{pll\_clk}$).
13. Optional: Set ROI via I²C interface (ROI_tl_x_hi/lo, ROI_tl_y, ROI_br_x_hi/lo, ROI_br_y registers).
14. Set the integration time via I²C (INTM_hi/lo, Int_len_hi/lo registers).
15. Do the counterpart for the frame buffer size adjustment, pointer initializations, DMA and INT settings on the application CPU.
16. Start the frame acquisition by using shutter trigger signal (SHUTTER = 1 or shutter_en = 1 via I²C).
17. Receive transmitted frames from TCMI interface to the external frame buffer on the application CPU.
18. Optional: Read the temperature sensor readout values via I²C
(Sum_Temp_tl_hi/lo, Sum_Temp_tr_hi/lo, Sum_Temp_bl_hi/lo, Sum_Temp_br_hi/lo registers).
19. Loop back to step 12 - 19.

Note: For corresponding I²C communication examples refer to chapter 5.4.7., Control commands.

### 7.2. Basic measurement mode setting

The basic measurement functions, as they are different modes of distance measurements and black & white imaging, are selected by the appropriate settings of the corresponding control registers. For the most often used applications, Table 26 lists these configurations.

Refer to the corresponding chapters for detailed explanations to each mode.

| Mode | | | Register setting | | | |
|---|---|---|---|---|---|---|
| Basic | Function | LSFR stages | MOD_Control (0x92) | MT_0_hi (0x22) | MT_0_lo (0x24) | PN_POLY (0x8C/0x8D) |
| TOF [1] | Sine 4x DCS [1] | --- [2] | 0x34 [1] | 0x34 | 0x00 | --- [2] |
| | Sine 2x DCS | --- [2] | 0x14 | 0x34 | 0x00 | --- [2] |
| | PN 4x DCS | 8 | 0x77 | 0x34 | 0x00 | Table 25 |
| | | 9 | 0x76 | 0x34 | 0x00 | Table 25 |
| | | 10 | 0x75 | 0x34 | 0x00 | Table 25 |
| | | 11 | 0x74 | 0x34 | 0x00 | Table 25 |
| | | 12 | 0x73 | 0x34 | 0x00 | Table 25 |
| | | 13 | 0x72 | 0x34 | 0x00 | Table 25 |
| | | 14 | 0x71 | 0x34 | 0x00 | Table 25 |
| | PN 2x DCS | 8 | 0x57 | 0x34 | 0x00 | Table 25 |
| | | 9 | 0x56 | 0x34 | 0x00 | Table 25 |
| | | 10 | 0x55 | 0x34 | 0x00 | Table 25 |
| | | 11 | 0x54 | 0x34 | 0x00 | Table 25 |
| | | 12 | 0x53 | 0x34 | 0x00 | Table 25 |
| | | 13 | 0x52 | 0x34 | 0x00 | Table 25 |
| | | 14 | 0x51 | 0x34 | 0x00 | Table 25 |
| Black & white | ambient only [3] | --- [2] | 0x00 | 0x14 | 0x26 | --- [2] |
| | ambient & non modulated LED/LD [4] | --- [2] | 0x00 | 0x14 | 0x16 | --- [2] |
| | ambient & modulated LED/LD [4] | --- [2] | 0x00 | 0x14 | 0x06 | --- [2] |

Table 26: Basic measurement mode setting

Notes:

[1] Default setting

[2] Setting is not applicable for this application.

[3] Black & white image passively illuminated by ambient-light only.

[4] Black & white image passively illuminated by ambient-light and actively illuminated by non modulated LED/LD.

[5] Black & white image passively illuminated by ambient-light and actively illuminated by modulated LED/LD.
Note: LED driver is always turned on. Take care that the LED driver and the epc660 chip does not exceed the maximum operating limits.

[6] For correct setting of DLL_measurement_rate, refer to chapter 5.5.2., DLL (Delay Locked Loop).

### 7.3. 3D TOF distance measurement flow

A final 3D TOF distance image will be done with different process steps according to Figure 57. Both interfaces of the epc660 are used: The $I^2C$ for configuration, mode selection and temperature reading (blue marked in the following figures) and the high-speed TCMI for reading the frame data (red marked in the following figures).

The sequence starts with the initialization of the epc660 registers with the necessary and correct configuration parameters. Next, the TOF measurement with the expected mode (sine, PN, 4x DCS, 2x DCS) will be performed. Depending of the application and the ambient conditions (ambient-light, changing temperature conditions), the TOF measurement needs some compensation. For the purpose of more accurate ambient-light compensation, a black & white measurement without illumination captures the background light level. Reading of the on-chip temperature sensors (from time to time) helps to compensate thermal influences caused by e.g. the LEDs, the optical filters and the sensor. After the rearrangement of the black & white image to the correct pixel orientation, the final 3D TOF distance image can be calculated with all for the application necessary compensations.

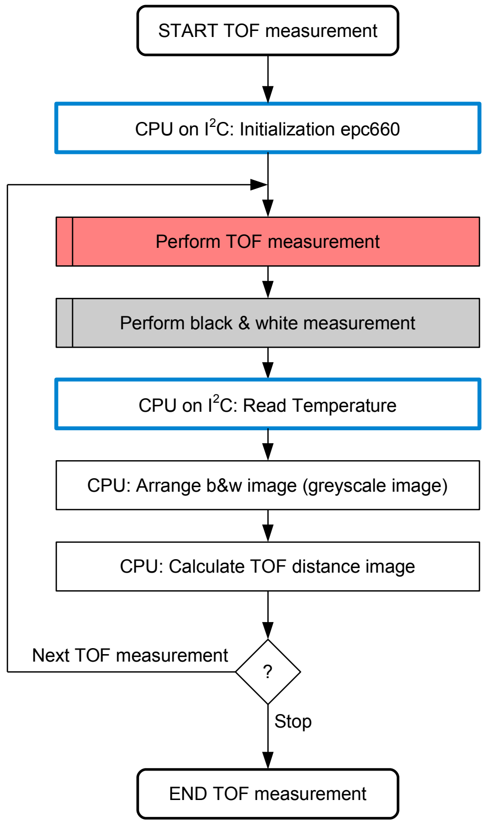



Figure 57: Generic 3D TOF distance measurement flow

The process flows for distance measurements and for black & white images are similar, see Figure 58. The main differences are the mode selection, see Table 26 and depending thereof the number of frames, which need to be read out during a process cycle. After mode setting, the cycle will be started by applying the SHUTTER signal. Once the SHUTTER is stimulated, the epc660 executes the measurement until the end of the sequence automatically. The application CPU has to follow accordingly by reading out the collected data until the end of sequence.

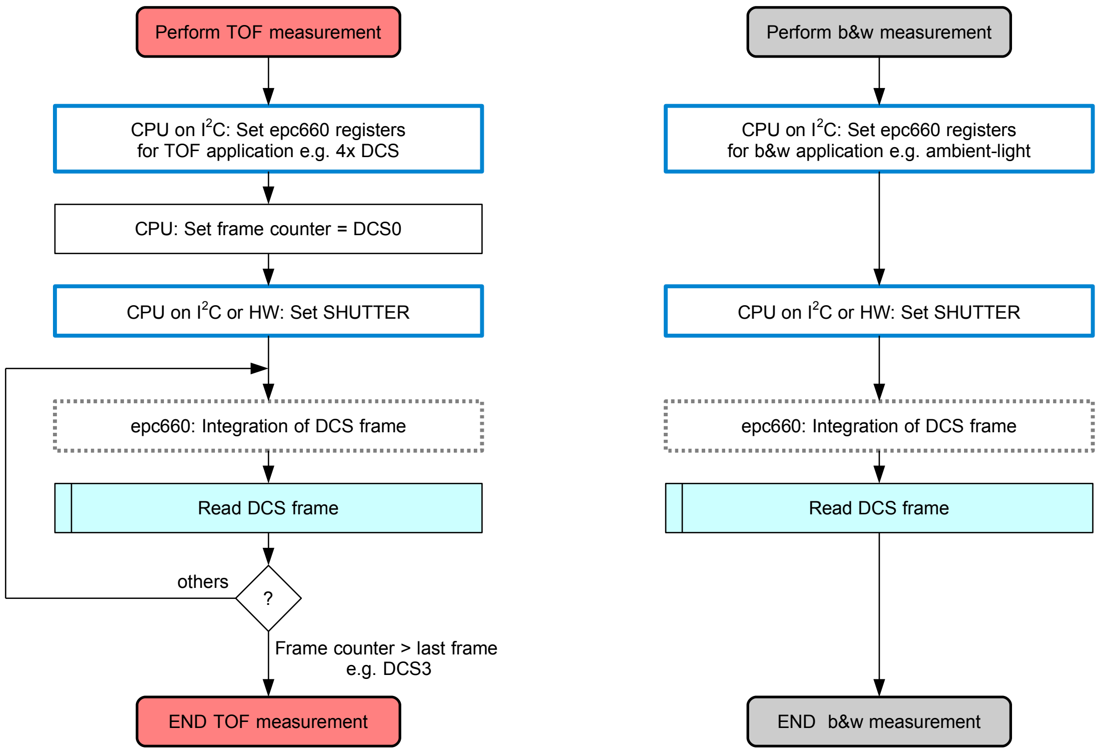



Figure 58: Generic sequences for the distance (TOF) and the black & white measurement

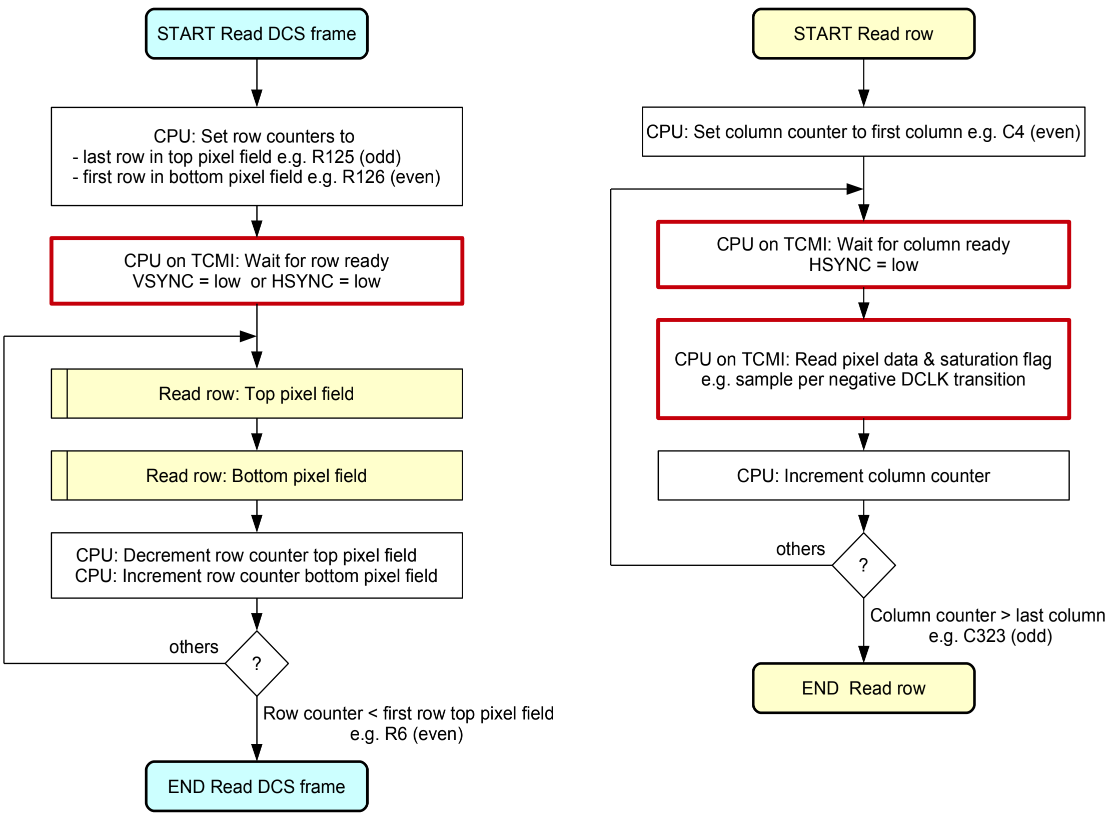



Figure 59: Generic sequences to readout frames and row by row

The generic procedures to readout frames or rows are independently of the selected modes. The application is driven only by the TCMI interface during these phases. To catch the beginning of the frame, the application CPU has to wait after the measurement start until the in -

tegration period is finished and the first frame data is available. The epc660 signals this by setting VSYNC and HSYNC active. Pixel data can be read DCLK by DCLK as long the HSYNC signal is active. Refer also to Figure 42 and Figure 43. The application has to take care of its own, to update synchronously all necessary frame, row and pixel readout counters of the application during the the whole measurement cycle.

Only if the validity and the necessary quality level of the measured data are given, the result of the distance is correct and reliable. The generic validity and quality of the data is independent of any correction or compensation algorithms. It is given by the complete epc660 camera system with chip, lens, illumination and environmental conditions.

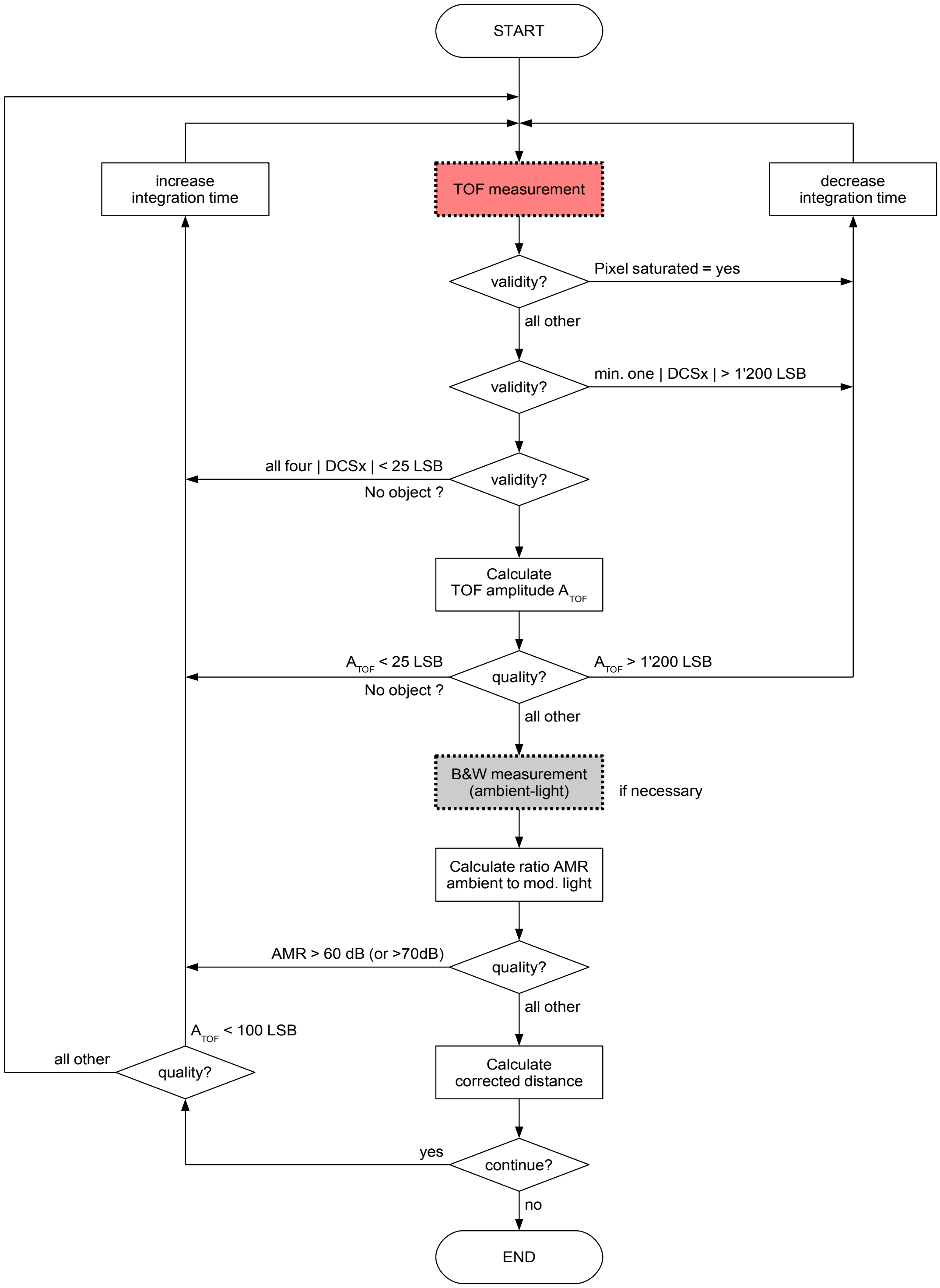



Figure 60: Generic validity and quality flow chart for a single pixel

Regarding possible correction and compensation algorithms refer to e.g. epc's application note AN10 or the Handbook epc600, which are available on the epc website.

### 7.4. Operating and unambiguity range versus time base setting

The epc660 3D TOF imager uses the time-of-flight principle. It is implemented with a repeating, continuous-mode modulation signal (sine or PN) during the integration phase (refer to chapter 6., Measurement Modes). Consequently, only signals returning within the maximum time slots can be detected at all (PN mode) or unambiguously (sine or PN mode).

The operating range is the maximum distance which corresponds to the maximum time-of-flight inside of one period of the used modulation: for sine, it is one period of $f_{LED}$ and for PN mode, it is the chip period $t_c$. Objects inside this area are detected unambiguously. For PN mode it means also, that objects above this distance and below the unambiguity distance are suppressed and cannot be detected (see Figure 56).

The unambiguity range defines the repetition distance, where objects outside of the targeted operating range can still be detected as far they are of very high reflectivity (remission). Strongly reflected signals outside of this range may therefore interfere with the measurement.

The operating range, the unambiguity distance, the time base for the integration time and the resolution of the distance signal are defined by the modulation clock mod_clk. This corresponds for the epc660 to a maximum default operating range of 7.5m @ mod_clk = 80MHz. It may be necessary depending on the application to adapt this parameters to other values. It can be done by a change of the mod_clk. Table 27 lists as an example some values of the modulation clock mod_clk in function of the operating range, the unambiguity distance, of the distance resolution and of the multiplier of the integration time base.

| TOF sine & PN | | | | | TOF sine | | TOF PN | |
|---|---|---|---|---|---|---|---|---|
| Operating range | Int. time multiplied by | Distance resolution [2] | Modulator clock | mod_clk _div | LED/LD mod. freq. | Unambiguity range | Chip period | Unambiguity range |
| | | | mod_clk | (0x85) | $f_{LED}$ | | $t_C$ | e.g. LSFR 8 |
| [m] | [#] | [cm] | [MHz] | [#] | [MHz] | [m] | [ns] | [m] |
| 7.50 [1] | 1 | 0.250 | 80.0 | 0 [1] | 20.000 | 7.50 | 50.0 | 1'912 |
| 15.00 | 2 | 0.500 | 40.0 | 1 | 10.000 | 15.00 | 100.0 | 3'825 |
| 30.00 | 4 | 1.000 | 20.0 | 3 | 5.000 | 30.00 | 200.0 | 7'650 |
| 60.00 | 8 | 2.000 | 10.0 | 7 | 2.500 | 60.00 | 400.0 | 15'300 |
| 120.00 | 16 | 4.000 | 5.0 | 15 | 1.250 | 120.00 | 800.0 | 30'600 |
| 240.00 | 32 | 8.000 | 2.5 | 31 | 0.625 | 240.00 | 1'600.0 | 61'200 |

Table 27: Operating range versus mod_clk

Notes:

[1] Default values

[2] The distance resolution is given for an operating range corresponding to 3'000 LSB.

### 7.5. Integration time setting

The integration time is the active frame acquisition period (see Figure 42). Specially for moving objects or cameras, this time should be very short to reduce motion blur effects as much as possible. The integration time together with the illumination intensity also defines the effective achievable distance, which the complete epc660 camera system can see.

The following Table 28 lists the possible and mostly common used integration times per measurement mode as well as the corresponding register settings.

| Integration time [5] | Remarks [1] | Register values | | | |
|---|---|---|---|---|---|
| | | INTM (0xA0/0xA1) | | Int_len (0xA2/0xA3) | |
| | | [DEC] | [HEX] | [DEC] | [HEX] |
| 100 ns | Sine & B&W only | 1d | 0x0001 | 7d | 0x0007 |
| 200 ns | | 1d | 0x0001 | 15d | 0x000F |
| 400 ns | | 1d | 0x0001 | 31d | 0x001F |
| 800 ns | | 1d | 0x0001 | 63d | 0x003F |
| 1.60 µs | | 1d | 0x0001 | 127d | 0x007F |
| 3.20 µs | | 1d | 0x0001 | 255d | 0x00FF |
| 6.40 µs | | 1d | 0x0001 | 511d | 0x01FF |
| 12.75 µs [5] | LSFR 8 | 1d | 0x0001 | 1'019d | 0x03FB |
| 25.55 µs | LSFR 9 | 1d | 0x0001 | 2'043d | 0x07FB |
| 51.15 µs | LSFR 10 | 1d | 0x0001 | 4'091d | 0x0FFB |
| 102.35 µs | LSFR 11 | 1d | 0x0001 | 8'187d | 0x1FFB |
| 204.75 µs | LSFR 12 | 1d | 0x0001 | 16'379d | 0x3FFB |
| 409.55 µs | LSFR 13 | 1d | 0x0001 | 32'763d | 0x7FFB |
| 819.15 µs | LSFR 14 | 1d | 0x0001 | 65'531d | 0xFFFB |
| 1.6383 ms | Multiples of LSFR 14 | 2d | 0x0002 | 65'531d | 0xFFFB |
| 3.2766 ms | | 4d | 0x0004 | 65'531d | 0xFFFB |
| 6.5532 ms | | 8d | 0x0008 | 65'531d | 0xFFFB |
| 13.1064 ms | | 16d | 0x0010 | 65'531d | 0xFFFB |
| 26.2128 ms | | 32d | 0x0020 | 65'531d | 0xFFFB |
| 52.4256 ms | | 64d | 0x0040 | 65'531d | 0xFFFB |
| 104.8512 ms | | 128d | 0x0080 | 65'531d | 0xFFFB |
| 209.7024 ms | | 256d | 0x0100 | 65'531d | 0xFFFB |
| 419.4048 ms | | 512d | 0x0200 | 65'531d | 0xFFFB |
| 838.8096 ms | | 1'024d | 0x0400 | 65'531d | 0xFFFB |
| 1.6776192 s | | 2'048d | 0x0800 | 65'531d | 0xFFFB |
| 3.3552384 s | | 4'096d | 0x1000 | 65'531d | 0xFFFB |
| 6.7104768 s | | 8'192d | 0x2000 | 65'531d | 0xFFFB |
| 13.4209536 s | | 16'384d | 0x4000 | 65'531d | 0xFFFB |
| 26.8419072 s | | 32'768d | 0x8000 | 65'531d | 0xFFFB |

Table 28: Typical TOF and B&W integration time settings for mod_clk = 80MHz

Notes:

[1] Sine & B&W only: These integration times cannot be used with PN mode. Next higher value is the double.
LSFR 8 … 14: These integration times are adjusted to the basic LSFR lengths (see Table 25). They follow the LSFR rules and do not exactly double for next higher value.
Multiples of LSFR 14: They are corresponding multiples of the integration time for LSFR 15. They follow the LSFR rules and do not exactly double for next higher value.

[2] Preferred integration time range for sine: 100 ns … 52.4256 ms, see also note 1.

[3] Preferred integration time range for PN mode: 12.75 µs … 52.4256 ms, see also note 1.

[4] Preferred integration time range for B&W mode: 100 ns … 52.4256 ms, see also note 1.

[5] Integration time default setting is 12.75µs.

### 7.6. Special mode setting

In this chapter, the user will find the register setting tables for using the special modes and will see its effects.
Detailed descriptions are given in the corresponding chapters of these modes; see chapter 5.6., Pixel Field.

#### 7.6.1. Binning, column/row reduction in basic measurement modes (Single MGX mode)

| **Ref.** | **Imager settings (Input)** | | | **Register value** | **Imager output** | | |
|---|---|---|---|---|---|---|---|
| **Figure** | **Binning [1] hor. / ver.** | **Column reduction x-axis: 2** | **Row reduction y-axis: 2, 4, 8** | **Resolution_ reduction (0x94)** | **Resolution x-y [2] [imager pixel]** | **Frame rate [fps]** | **Frame size [3] [kbytes]** |
| 35 | --- | --- | --- | 0x00 | 320 x 240 | 262 | 150 |
| 35 | --- | --- | 2 | 0x04 | 320 x 120 | 524 | 75 |
| 35 | --- | --- | 4 | 0x08 | 320 x 60 | 1'048 | 37.5 |
| 35 | --- | --- | 8 | 0x0C | 320 x 30 | 2'096 | 18.75 |
| --- | --- | 2 | --- | 0x01 | 160 x 240 | 524 | 75 |
| --- | --- | 2 | 2 | 0x05 | 160 x 120 | 1'048 | 37.5 |
| --- | --- | 2 | 4 | 0x09 | 160 x 60 | 2'096 | 18.75 |
| --- | --- | 2 | 8 | 0x0D | 160 x 30 | 4'192 | 9.375 |
| 36 | horizontal | 2 | --- | 0x11 | 160 x 240 | 524 | 75 |
| 36 | horizontal | 2 | 2 | 0x15 | 160 x 120 | 1'048 | 37.5 |
| 36 | horizontal | 2 | 4 | 0x19 | 160 x 60 | 2'096 | 18.75 |
| 36 | horizontal | 2 | 8 | 0x1D | 160 x 30 | 4'192 | 9.375 |
| 37 | vertical | --- | 2 | 0x24 | 320 x 120 | 524 | 75 |
| 37 | vertical | --- | 4 | 0x28 | 320 x 60 | 1'048 | 37.5 |
| 37 | vertical | --- | 8 | 0x2C | 320 x 30 | 2'096 | 18.75 |
| 38 | both | 2 | 2 | 0x35 | 160 x 120 | 1'048 | 37.5 |
| 38 | both | 2 | 4 | 0x39 | 160 x 60 | 2'096 | 18.75 |
| 38 | both | 2 | 8 | 0x3D | 160 x 30 | 4'192 | 9.375 |

Table 29: Binning, column/row reduction in basic measurement modes (Single MGX mode, default: QVGA)

Notes:

[1] Horizontal or vertical binning correspond to a 2x increased sensitivity. Binning on both axis corresponds to a 4x increased sensitivity.

[2] Readout has to follow the effective numbers and order of columns and rows.

[3] Frame size is based on 12 bit DATA[11:0] + 1 bit SAT flag per sample packed in 2 Bytes to store in the ext. frame buffer.

### 7.6.2. Binning, column/row reduction in enhanced measurement modes (Dual MGX mode)

| Ref. | Imager settings (Input) | | | Register value | Imager output | | |
|---|---|---|---|---|---|---|---|
| Figure | Binning [1] hor. / ver. | Column reduction x-axis: 2 | Row reduction y-axis: 2, 4, 8 | Resolution_ reduction (0x94) | Resolution x-y [2] [imager pixel] | Frame rate [fps] | Frame size [3] [kbytes] |
| 39 | --- | --- | --- | 0x80 | 2 x 320 x 120 | 262 | 150 |
| 39 | --- | --- | 2 | 0x84 | 2 x 320 x 60 | 524 | 75 |
| 39 | --- | --- | 4 | 0x88 | 2 x 320 x 30 | 1'048 | 37.5 |
| --- | --- | 2 | --- | 0x81 | 2 x 160 x 120 | 524 | 75 |
| --- | --- | 2 | 2 | 0x85 | 2 x 160 x 60 | 1'048 | 37.5 |
| --- | --- | 2 | 4 | 0x89 | 2 x 160 x 30 | 2'096 | 18.75 |

Table 30: Binning, column/row reduction in enhanced measurement modes: Dual MGX mode, default: QVGA

Notes:

[1] No binning available in this mode.

[2] Readout has to follow the effective numbers and order of columns and rows.

[3] Frame size is based on 12 bit DATA[11:0] + 1 bit SAT flag per sample packed in 2 Bytes to store in the ext. frame buffer.

**TOF dual DCS acquisition at same integration time (motion blur reduction)**

| Mode | | | Register setting | | |
|---|---|---|---|---|---|
| Basic | Function | LSFR stages | MOD_Control (0x92) | MT_1_hi (0x25) | PN_POLY (0x8C/0x8D) |
| TOF | Sine 4x DCS | --- [1] | --- [1] | --- [1] | --- [1] |
| | Sine 2x DCS [2] | --- [1] | 0x14 | 0x3E [2] | --- [1] |
| | PN 4x DCS | --- [1] | --- [1] | --- [1] | --- [1] |
| | PN 2x DCS [2] | 8 | 0x57 | 0x3E [2] | Table 25 |
| | | 9 | 0x56 | 0x3E [2] | Table 25 |
| | | 10 | 0x55 | 0x3E [2] | Table 25 |
| | | 11 | 0x54 | 0x3E [2] | Table 25 |
| | | 12 | 0x53 | 0x3E [2] | Table 25 |
| | | 13 | 0x52 | 0x3E [2] | Table 25 |
| | | 14 | 0x51 | 0x3E [2] | Table 25 |
| Black & white | --- [1] | | | | |

Table 31: Basic measurement mode setting

Notes:

[1] Setting is not applicable for this application.

[2] Output is effectively 4x DCS in 2 DCS-frames. MT_1_hi register default value: 0x31

**TOF and b&w single DCS acquisition with 2 different integration times (High dynamic range)**

- This mode needs the following basic setting of the MT registers:
  MT_0_hi = 0x30, MT_1_hi = 0x35, MT_2_hi = 0x3A, MT_3_hi = 0x3F.
- Reset the MT registers to the default values after leaving this mode:
  MT_0_hi = 0x34, MT_1_hi = 0x31, MT_2_hi = 0x3E, MT_3_hi = 0x3B.

| **Mode** | | | **Register setting** | | | |
|---|---|---|---|---|---|---|
| **Basic** | **Function** [5] | **LSFR stages** | **MOD_Control (0x92)** | **MT_0_hi (0x22)** | **MT_0_lo (0x24)** | **PN_POLY (0x8C/0x8D)** |
| TOF | Sine 4x DCS | --- [1] | 0x3C [1] | 0x30 | 0x00 | --- [1] |
| | Sine 2x DCS | --- [1] | 0x1C | 0x30 | 0x00 | --- [1] |
| | PN 4x DCS | 8 | 0x7F | 0x30 | 0x00 | Table 25 |
| | | 9 | 0x7E | 0x30 | 0x00 | Table 25 |
| | | 10 | 0x7D | 0x30 | 0x00 | Table 25 |
| | | 11 | 0x7C | 0x30 | 0x00 | Table 25 |
| | | 12 | 0x7B | 0x30 | 0x00 | Table 25 |
| | | 13 | 0x7A | 0x30 | 0x00 | Table 25 |
| | | 14 | 0x79 | 0x30 | 0x00 | Table 25 |
| | PN 2x DCS | 8 | 0x5F | 0x30 | 0x00 | Table 25 |
| | | 9 | 0x5E | 0x30 | 0x00 | Table 25 |
| | | 10 | 0x5D | 0x30 | 0x00 | Table 25 |
| | | 11 | 0x5C | 0x30 | 0x00 | Table 25 |
| | | 12 | 0x5B | 0x30 | 0x00 | Table 25 |
| | | 13 | 0x5A | 0x30 | 0x00 | Table 25 |
| | | 14 | 0x59 | 0x30 | 0x00 | Table 25 |
| Black & white | ambient only [2] | --- [1] | 0x08 | 0x10 | 0x26 | --- [1] |
| | ambient & non modulated LED/LD [3] | --- [1] | 0x08 | 0x10 | 0x16 | --- [1] |
| | ambient & modulated LED/LD [4] | --- [1] | 0x08 | 0x10 | 0x06 | --- [1] |

Table 32: Measurement mode setting for high dynamic range

Notes:

[1] Setting is not applicable for this application.

[2] Black & white image passively illuminated by ambient-light only.

[3] Black & white image passively illuminated by ambient-light and actively illuminated by non modulated LED/LD.

[4] Black & white image passively illuminated by ambient-light and actively illuminated by modulated LED/LD.
Note: LED driver is always turned on. Take care that the LED driver and the epc660 chip does not exceed the maximum operating limits.

[5] Output is effectively 2 DCS frames with different integration times in each 1 readout frame.

### 7.6.3. ROI setting (region of interest)

Note: For the emulation of the epc635 refer to the application section: chapter 7.6.4., epc635 emulation.

| Ref. | Imager | | ROI setting | Register setting top left | | Register setting bottom right | |
|---|---|---|---|---|---|---|---|
| Figure | Type | Resolution x-y | Resolution x-y | X (0x96/0x97) | Y (0x9A) | X (0x98/0x99) | Y (0x9B) |
| | | [imager pixel] | [imager pixel] | [DEC] | [DEC] | [DEC] | [DEC] |
| 41 | QVGA (Default) | 320 x 240 | 320 x 120 | 4 | 6 | 323 | 125 |
| 45 | Half-QQVGA | 160 x 60 | 160 x 30 | 84 | 96 | 243 | 125 |

Table 33: ROI setting for Half-QQVGA

| Ref. | Imager settings (Input) | | | Register value | Imager output | | |
|---|---|---|---|---|---|---|---|
| Figure | Binning [1] hor. / ver. | Column reduction x-axis: 2 | Row reduction y-axis: 2, 4, 8 | Resolution_ reduction (0x94) | Resolution x-y [2] [imager pixel] | Frame rate [fps] | Frame size [3] [kbytes] |
| | **Single MGX mode** | | | | | | |
| 35 | --- | --- | --- | 0x08 | 160 x 60 | 1'048 | 18.75 |
| 35 | --- | --- | 2 | 0x0C | 160 x 30 | 2'096 | 9.375 |
| --- | --- | 2 | --- | 0x09 | 80 x 60 | 2'096 | 9.375 |
| --- | --- | 2 | 2 | 0x0D | 80 x 30 | 4'192 | 4.6875 |
| 36 | horizontal | 2 | --- | 0x19 | 80 x 60 | 2'096 | 9.375 |
| 36 | horizontal | 2 | 2 | 0x1D | 80 x 30 | 4'192 | 4.6875 |
| 37 | vertical | --- | 2 | 0x2C | 160 x 30 | 2'096 | 9.375 |
| 38 | both | 2 | 2 | 0x3D | 80 x 30 | 4'192 | 4.6875 |
| | **Dual MGX mode** | | | | | | |
| 39 | --- | --- | --- | 0x88 | 2 x 160 x 30 | 1'048 | 37.5 |
| --- | --- | 2 | --- | 0x89 | 2 x 80 x 30 | 2'096 | 18.75 |

Table 34: Binning, column/row reduction for ROI setting for Half-QQVGA and MGX modes

Notes:

[1] Horizontal or vertical binning correspond to a 2x increased sensitivity. Binning on both axis corresponds to a 4x increased sensitivity.

[2] Readout has to follow the effective numbers and order of columns and rows.

[3] Frame size is based on 12 bit DATA[11:0] + 1 bit SAT flag per sample packed in 2 Bytes to store in the ext. frame buffer.

### 7.6.4. epc635 emulation

epc660 can 100% emulate the epc635 (160x60 pixel) in terms of frame rate, resolution and data transfer by reprogramming the chip. The reprogramming takes place by downloading the new program segments via I²C.

Figure 61 and Figure 62 show the mechanical and optical compatibilities between both chip types.

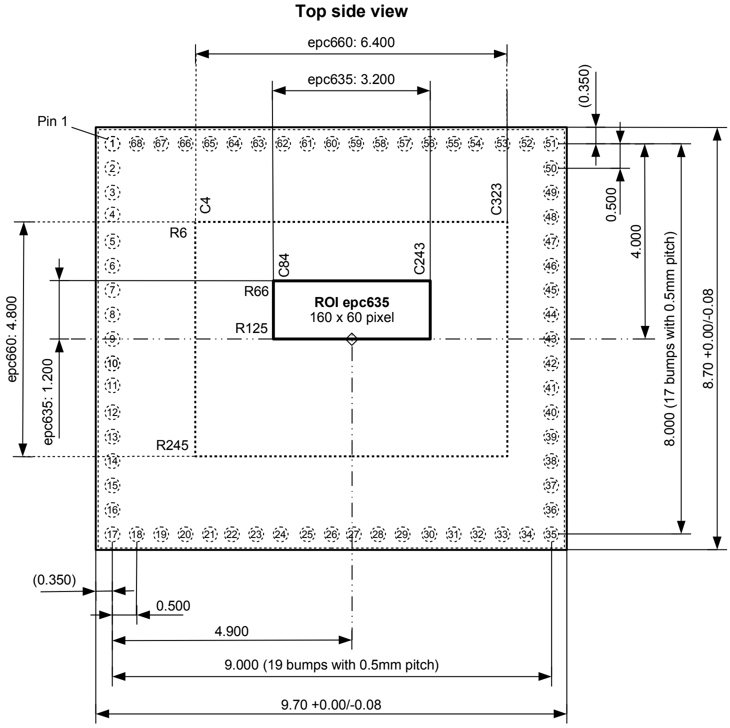



Figure 61: epc660: ROI setting for epc635 emulation

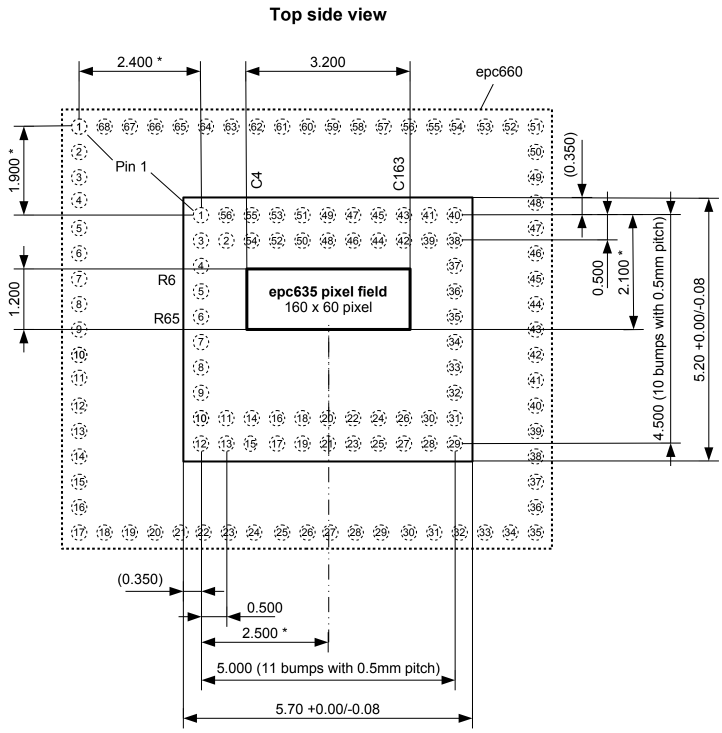



Figure 62: Mechanical dimensions epc635 versus epc660, pin1

* Note: This dimensions may slightly change in the final design of the epc635.

Change to epc635 mode:
1. Start up epc660 chip
2. Wait until the chip is in READY state.
3. Download the following program segments by I²C for reprogramming the epc660 to get into the epc635 emulation mode.
4. The chip is now ready to be used as epc635.
   The commands are the same as for the epc660, except the ROI is limited to the upper half pixel-field.

5. Exit epc635 mode: Power down or RESET sets the chip back into the epc660 mode.

**IMPORTANT:**
epc635 mode is reading out the upper part of the pixel field only. It means, the data readout starts at the last row (125) and proceeds in a continuous way upwards until the first row (66) without any swaps between upper and lower pixel-field rows.

**Program segments for reprogramming the epc660 by I²C download**

Note: The actual download code for each chip version can be found in the document "Operating_Instruction_epc660-XXX-Vx.x"

After the program download, the chip will be operated by the regular instructions of the epc660. Special settings will deviate according to the Table 35 and Table 36.

| Ref. | Imager | | ROI setting | Register setting top left | | Register setting bottom right | |
|---|---|---|---|---|---|---|---|
| Figure | Type | Resolution x-y | Resolution x-y | X (0x96/0x97) | Y (0x9A) | X (0x98/0x99) | Y (0x9B) |
| | | [imager pixel] | [imager pixel] | [DEC] | [DEC] | [DEC] | [DEC] |
| 41 | epc660: QVGA | 320 x 240 | 320 x 120 | 4 | 6 | 323 | 125 |
| 45 | epc635: Half-QQVGA | 160 x 60 | 160 x 60 | 84 | 66 | 243 | 125 |

Table 35: ROI setting for epc635 emulation

| Ref. | Imager settings (Input) | | | Register value | Imager output | | |
|---|---|---|---|---|---|---|---|
| Figure | Binning [1] hor. / ver. | Column reduction x-axis: 2 | Row reduction y-axis: 2, 4, 8 | Resolution_ reduction (0x94) | Resolution x-y [2] [imager pixel] | Frame rate [4] [fps] | Frame size [3] [kbytes] |
| | **Single MGX mode** | | | | | | |
| 35 | --- | --- | --- | 0x08 | 160 x 60 | 524 | 18.75 |
| 35 | --- | --- | 2 | 0x0C | 160 x 30 | 1'048 | 9.375 |
| --- | --- | 2 | --- | 0x09 | 80 x 60 | 1'048 | 9.375 |
| --- | --- | 2 | 2 | 0x0D | 80 x 30 | 2'096 | 4.6875 |
| 36 | horizontal | 2 | --- | 0x19 | 80 x 60 | 1'048 | 9.375 |
| 36 | horizontal | 2 | 2 | 0x1D | 80 x 30 | 2'096 | 4.6875 |
| 37 | vertical | --- | 2 | 0x2C | 160 x 30 | 1'048 | 9.375 |
| 38 | both | 2 | 2 | 0x3D | 80 x 30 | 2'096 | 4.6875 |
| | **Dual MGX mode** | | | | | | |
| 39 | --- | --- | --- | 0x88 | 2 x 160 x 30 | 524 | 37.5 |
| --- | --- | 2 | --- | 0x89 | 2 x 80 x 30 | 1'048 | 18.75 |

Table 36: Binning, column/row reduction for ROI setting: epc635 emulation

Notes:

[1] Horizontal or vertical binning correspond to a 2x increased sensitivity. Binning on both axis corresponds to a 4x increased sensitivity.

[2] Readout has to follow the effective numbers and order of columns and rows.

[3] Frame size is based on 12 bit DATA[11:0] + 1 bit SAT flag per sample packed in 2 Bytes to store in the ext. frame buffer.

[4] The frame rate is valid for 1µs integration time.

### 7.7. Power saving options

The epc660 has different power consumption depending on its activity level, see chapter 5.9., Power consumption levels. For power critical applications e.g. battery powered systems, it is possible to enforce the epc660 to go in so-called power saving states.

The different power saving modes and their description are listed in Table 37.

| Operating mode | Power | Time | | Description |
|---|---|---|---|---|
| | | Shutdown | Recovery | |
| | [mW] | [ms] | [ms] | |
| INTEGRATION | | | | Maximum average power consumption during regular 3D TOF operation |
| READY (Idle) | | | | Minimum average power consumption during regular 3D TOF operation |
| SW POWER DOWN | | | | Note: Only during regular READY mode possible.<br>Power consumption during unnecessary functions where switched-off off by register commands. |
| HW POWER DOWN | | | | Power consumption during Pin 55 $\overline{\text{RESET}}$ = 0. |

Table 37: Power modes

Table 38 is a help for the electronic designer and list the current consumption versus the power saving level and per supply level.

| Current | Description | Mode | | | | Units |
|---|---|---|---|---|---|---|
| | | INTEGRATION | READY | SW POWER DOWN | HW POWER DOWN | |
| $I_{VDD}$ | Digital supply current | 30 | | | | mA |
| $I_{VDDPLL}$ | PLL supply current | 6 | | | | mA |
| $I_{VDDIO}$ | IO supply current | 40 | | | | mA |
| $I_{VDDA}$ | Analog supply current | 250 | | | | mA |
| $I_{VDDPXM}$ | Pixel analog 1 supply current | 1 | | | | mA |
| $I_{VDDPXH}$ | Pixel analog 2 supply current | | | | | mA |
| $I_{VSEAL}$ | Sealring supply current | 2 | | | | mA |
| $I_{VIR}$ | Isolation supply current | | | | | mA |
| $I_{VBS}$ | Bias supply current | | | | | mA |

Table 38: Current consumption during power modes

Is the epc660 during regular operation in the READY state, the SW power down mode can be activated by the following command sequence listed Table 39.

| No. | Register | | | Description |
|---|---|---|---|---|
| | Name | Address | Value | |
| 1 | **Regular 3D TOF operation** | | | |
| 2 | Power_Control (analog) | 0xA5 | 0x00 | Switch off of unnecessary supplies |
| 3 | CLK_enables | 0x80 | 0x00 | Switch off of unnecessary clocks |
| 4 | CFG_Mode_Control | 0x7D | 0x14 | Switch system clock to XTAL clock |
| 5 | CFG_Mode_Control | 0x7D | 0x10 | Switch off PLL |
| 6 | **SW POWER DOWN** | | | |
| 7 | CFG_Mode_Control | 0x7D | 0x14 | Switch on PLL |
| 8 | **Wait > 32µs** | | | **Wait until PLL stable** |
| 9 | CFG_Mode_Control | 0x7D | 0x04 | Switch system clock to PLL |
| 10 | CLK_enables | 0x80 | 0x3F | Switch on the clocks again |
| 11 | Power_Control (analog) | 0xA5 | 0x07 | Switch on the supplies again |
| 12 | **Wait until supplies are stable** | | | |
| 13 | **Regular 3D TOF operation** | | | |

Table 39: SW POWER DOWN sequence

### 7.8. Rolling DCS frames

In special applications, it is possible to use all the time the same integration time in continuous distance measurement mode without any black & white images for ambient-light compensation. Such a set-up allows enhancing the distance measurement rate by a factor of 4 by using rolling DCS frames.

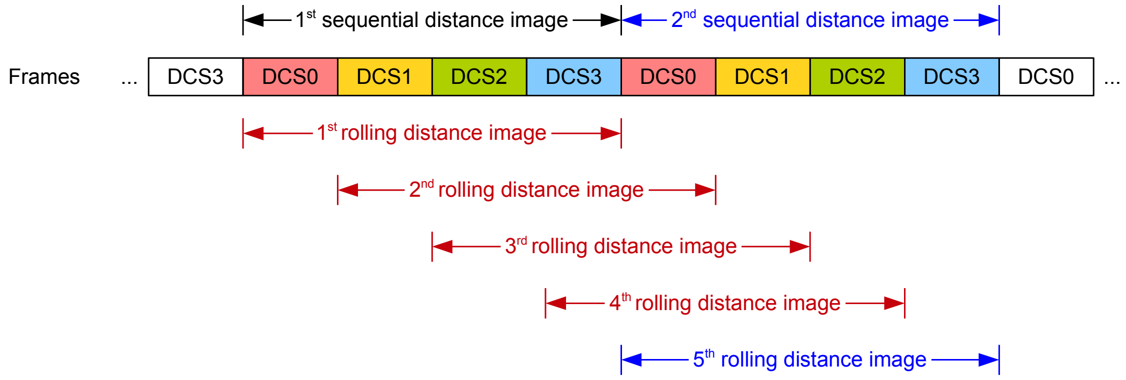



Figure 63: Rolling DCS frames

As shown in Figure 63, the algorithm performs with each new DCS frame a new distance calculation based on the new and last three DCS frames.

### 7.9. External modulation MODCLK

The epc660 has for enhanced user applications also the possibility to bring an external modulation clock to the chip. The optional MOD-CLK input can be used for this to inject in parallel a user controlled/modulated clock for both the LED driver and the pixel demodulator, see Figure 64.

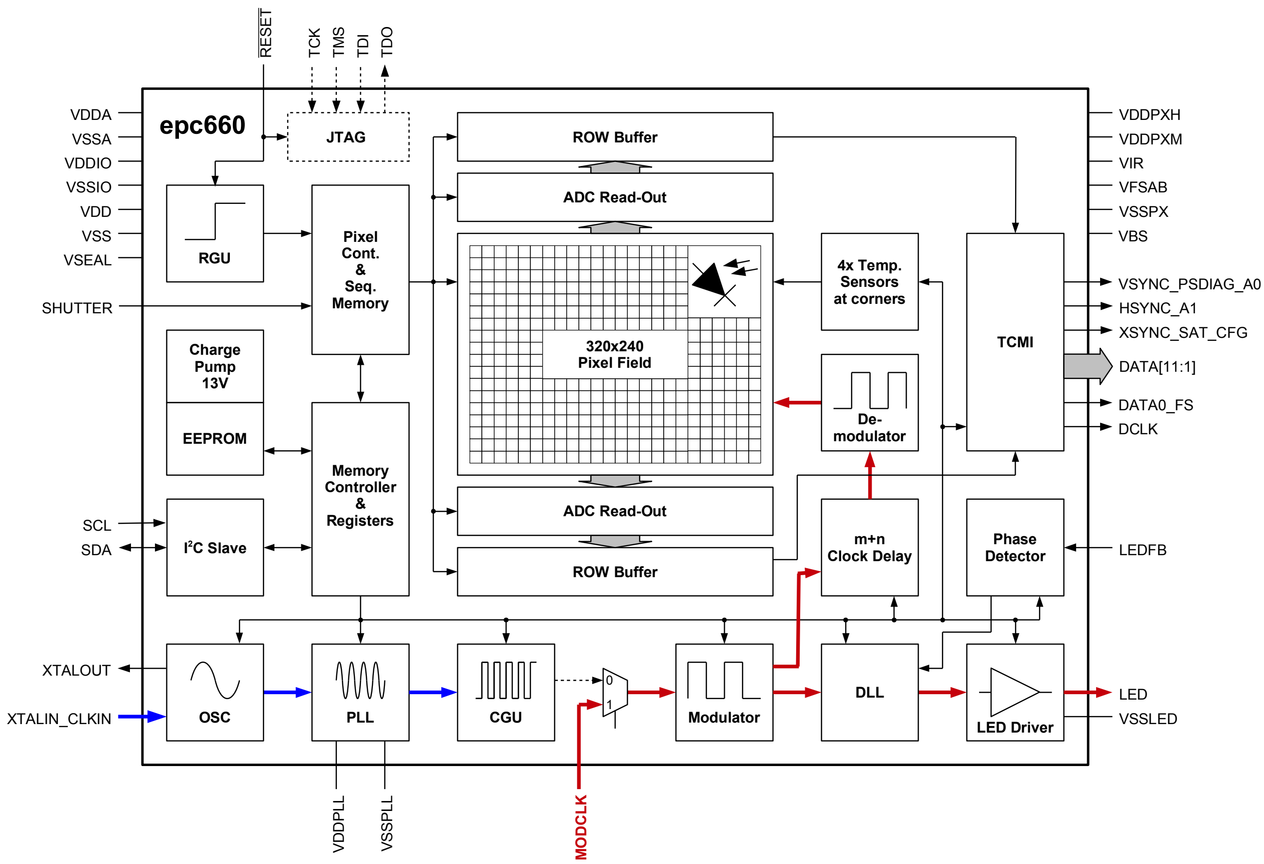



Figure 64: The MODCLK signal flow (red marked)

The external MODCLK can be used e.g. in concepts for reliable multi camera applications. It allows to use e.g. frequency-division multiple access (FDMA). In corresponding literature, the details of these concepts are explained in detail.

The user is free to apply any digital waveform up to 80MHz during frame acquisition as external MODCLK signal. Even more, he is also free to use modulations like pseudo-random edge jitter, dithering, etc.

The signal from the MODCLK pin is used as far in the register CLK_enables is set to mod_clk_sel = 1, see section 8.7.1.

**IMPORTANT:**

- The external modulation clock replaces the internal generated modulation clock, refer also to Table 27. Based on this clock, the modulator generates the modulation sequence according chapter 6., Measurement Modes and the setting in the MOD_Control register.
- Set DLL_measurement_rate = 0 for switching off the automatic DLL synchronization.

### 7.9.1. Using the DLL with an external modulation clock

Here, the modulator uses the ext_mod_clk signal from the MODCLK pin for both LED modulation and pixel demodulation functions when mod_clk_sel = 1. The user application is free to apply any digital waveform up to 80MHz during frame acquisition as ext_mod_clk signal. During the locking operation, this clock must be very stable, jitter-free, continuously running at least during the DLL pre-synchronization time and DLL lock time period. Is this period over, the user can switch back to his own modulation schemes: pseudo-random edge jitter, dithering, etc.

There are two possible ways to achieve this:

- Stop jitter/dithering and supply a clean clock on the MODCLK pin. The clock frequency for DLL synchronization needs to be the same as later on used during modulation period. Enable the DLL by setting dll_bypass = 0. Set the DLL_measurement_rate = 1. Do a 1x DCS frame acquisition. After the frame data-out is completed, set DLL_measurement_rate = 0. This will freeze the last measured delay in the DLL. Enable jitter/dithering on the MODCLK pin and continue the normal frame acquisition.
- When a clean clock cannot be provided on the MODCLK pin, the application need to switch first to internal PLL source by setting mod_clk_sel = 0. It is followed by repeating the same procedure as above. Before starting the normal frame acquisition, set mod_clk_sel = 1 for switching back to the modulation clock provided on the MODCLK pin.

### 7.9.2. DLL manual locking operation

DLL can also be operated manually via I²C communication by assigning externally a delay value to the DLL control registers for bringing it into a locking condition. The procedure follows by single register access steps in a loop by software the mechanism of the automatic locking, see chapter 5.5.2., DLL (Delay Locked Loop).

The Figure 27 shows the two paths which must be brought in synchronization: The signal propagation time of the red coloured path depends, among others, on the registers Demodulation_delays and Dist_offset. The mgx_del_sel value of the Demodulation_delays register will add a delay to this path if required. The purpose is, to make the propagation delay of this path longer than the blue path (see Figure 28). The Dist_offset register is application dependent and will be neglected in this section. On the other side, the propagation time of the blue coloured path is mainly defined by the external LED/LD circuit until the LEDFB pin. The DLL compares the feedback signals from both paths and provides status information in the DLL_status register. The delay lines of the DLL must now be adjusted by applying values to the registers DLL_coarse_ctrl_ext and DLL_fine_ctrl_ext_* until the DLL locks (see Table 41, DLL manual control registers).

Figure 65 provides a flowchart for manually DLL locking. The software has to determine the register settings for the DLL locking condition. Note, this is a straightforward control algorithm to explain the DLL manually locking behaviour. Different algorithms can be implemented depending on the applications.

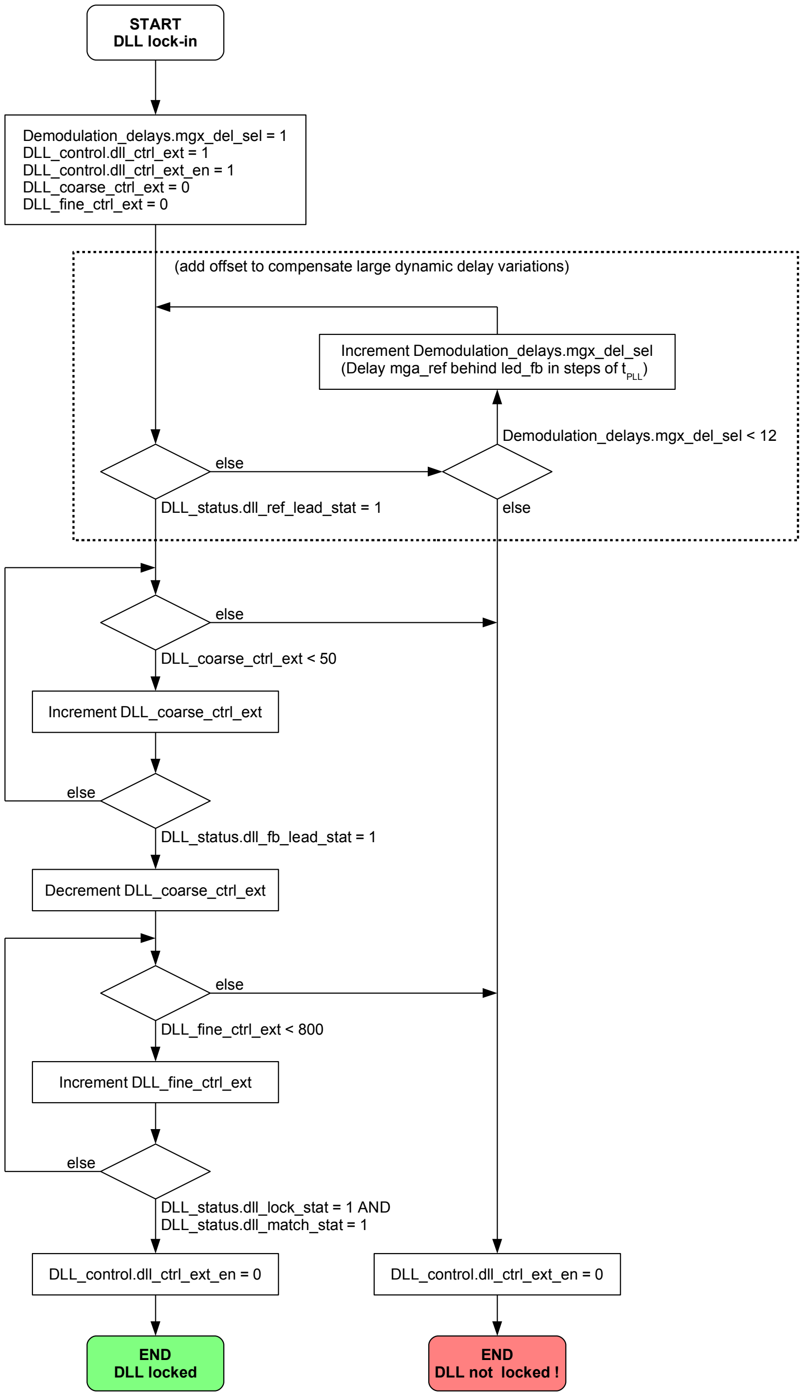



Figure 65: Flowchart for DLL manual locking

### 7.10. Calculation of PN mode polynomials (m-sequences)

The following Octave script can be used for calculating PN mode polynomials (m-sequences), which can be later on programmed into the PN_POLY_hi and PN_POLY_lo registers.

Following packages / toolboxes must be installed prior to run the Octave script:

| Package Name | Version |
|---|---|
| communications * | 1.1.1 |
| control * | 2.6.0 |
| general * | 1.3.2 |
| miscellaneous * | 1.0.9 |
| optim * | 1.0.12 |
| signal * | 1.2.2 |
| specfun * | 1.0.9 |

Table 40: Octave toolboxes

**Octave script pn_poly.m:**

```
% Load all packages
pkg load all

% Calculating for LFSR stages from 8 to 15
for i = 8 : 15
  printf("\n\n\n\nLSFR stage: %d\n", i)

  % Calculating primitive polynoms. Remove 'nodisplay' if you want to see
  % polynomials in a different form.
  %primitive_poly = primpoly(i, 'all');
  primitive_poly = primpoly(i, 'all', 'nodisplay');

  % Getting number of primitive polynomials. Result was written in the form of
  % a matrix with 1 row. We need to get the row length.
  num_of_poly = size(primitive_poly, 2);
  printf("num_of_poly: %d\n", num_of_poly)

  % Length of the PN sequence
  pn_len = 2^i - 1;
  printf("pn_len: %d\n", pn_len)

  % Print polynomials as number
  printf("\nPolynomials in the number form:\n")
  for j = 1 : num_of_poly
    %printf("%d\n", primitive_poly(j))    % decimal
    printf("0x%0x\n", primitive_poly(j))  % hex
  end
  printf("\n\n")
end

printf("Done.\n\n")
```

## 7.11. epc660 Card Edge Connector Carrier

### 7.11.1. Schematics

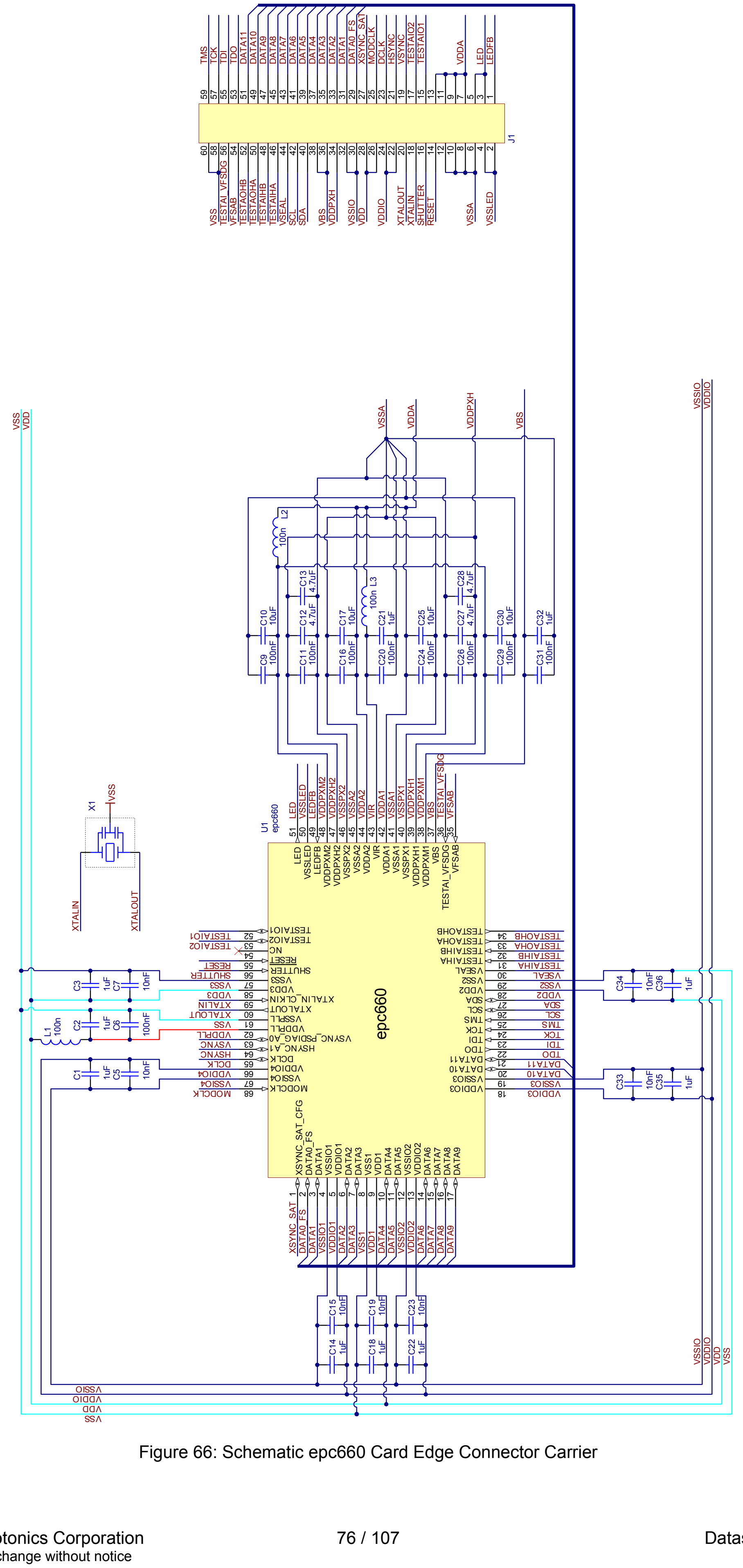


Figure 66: Schematic epc660 Card Edge Connector Carrier

### 7.11.2. Board layout and assembly

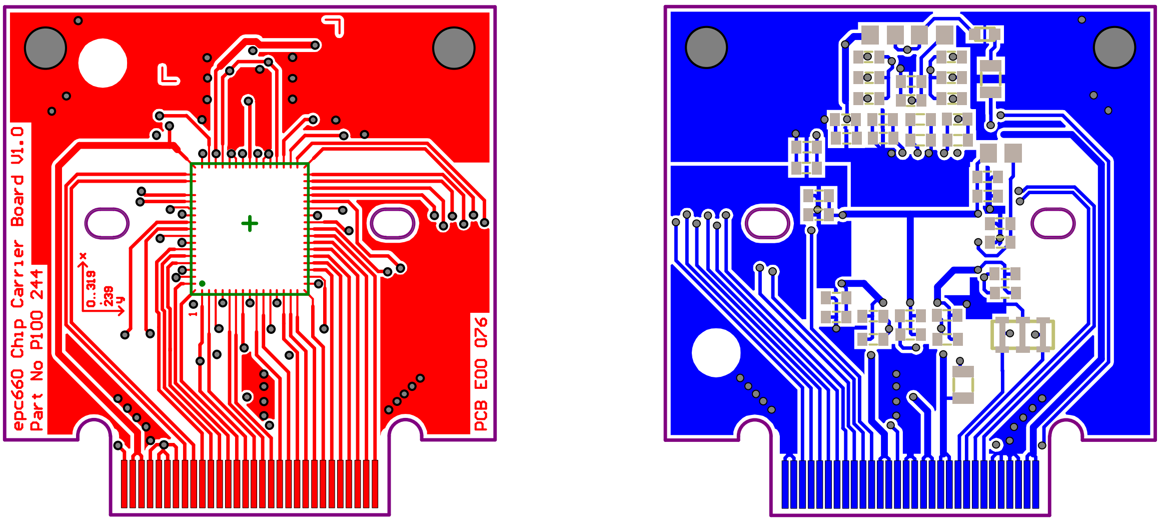



Figure 67: epc660 Card Edge Connector Carrier: Layout top and bottom

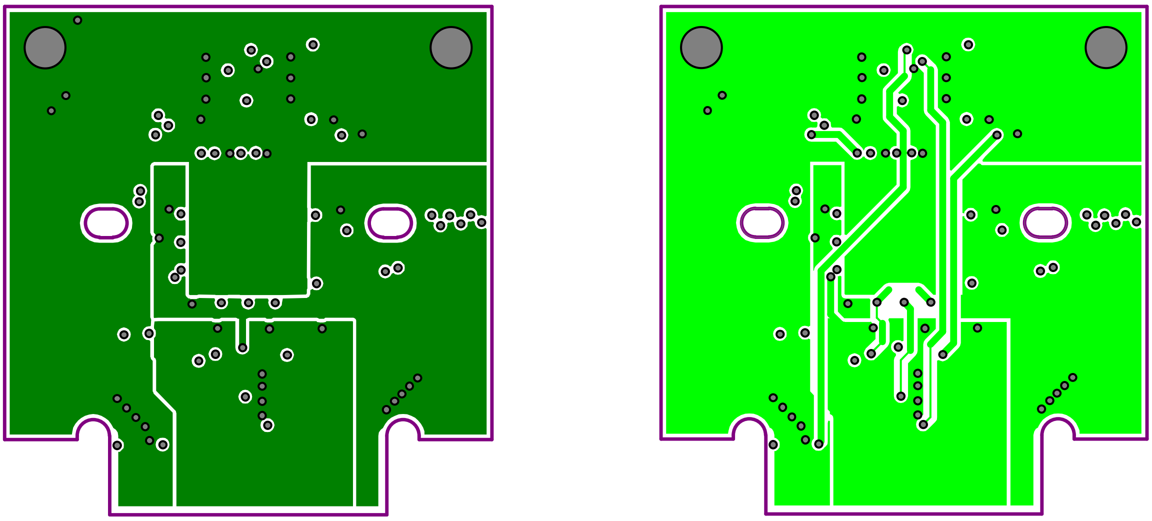


Figure 68: epc660 Card Edge Connector Carrier: Layout middle top and bottom

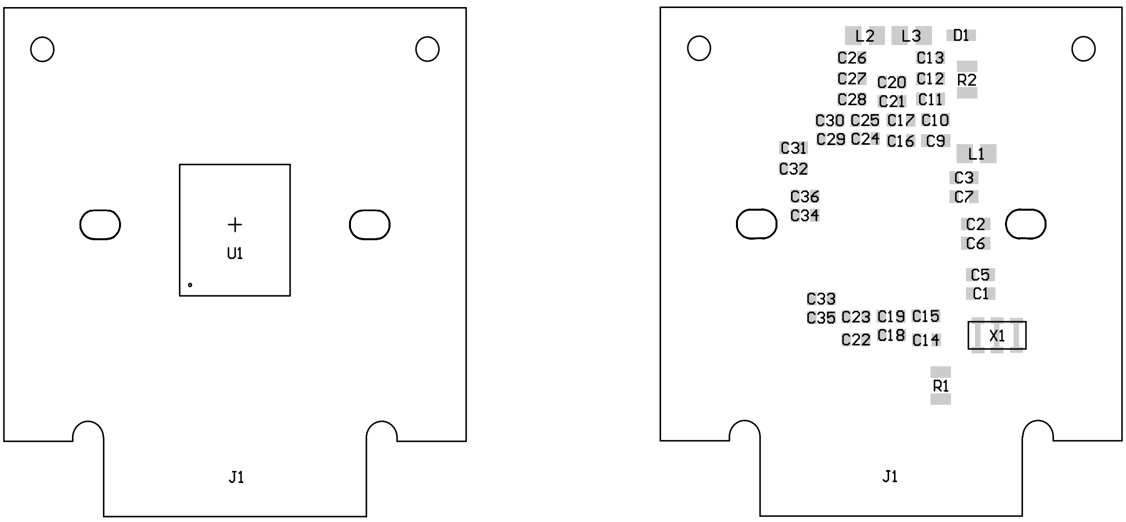



Figure 69: epc660 Card Edge Connector Carrier: Assembly top and bottom

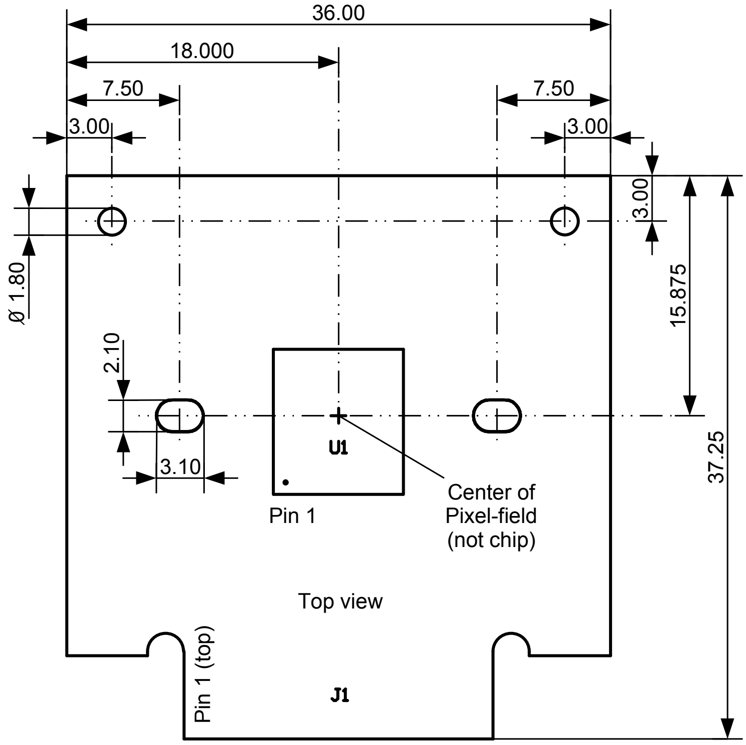



Figure 70: epc660 Card Edge Connector Carrier: Dimensions

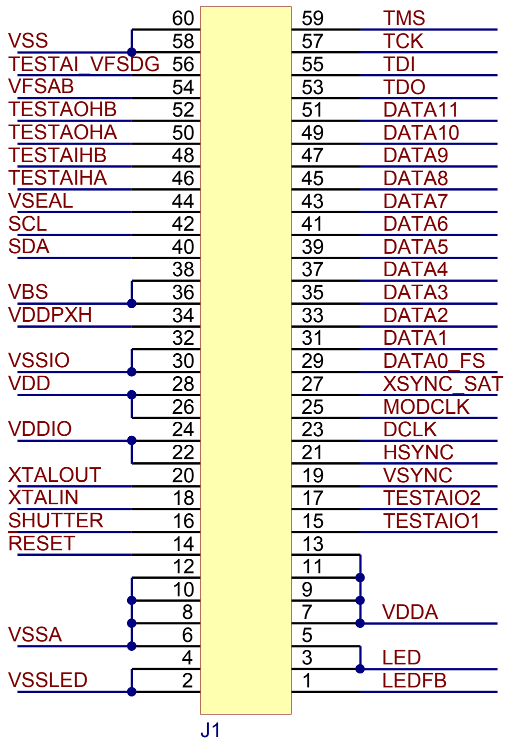



Figure 71: Pin table card connector

Figure 72 and Figure 73 show possible card connectors for interfacing the Card Edge Connector Carrier with the user's application board e.g. SAMTEC MEC6-130-02-L-DV-A / -RA1

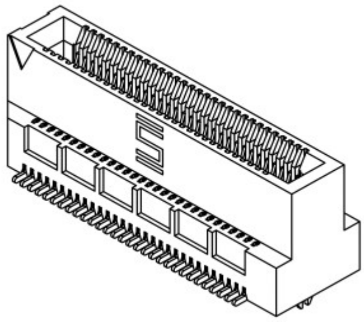


Figure 72: Vertical mount mini-edge card connector

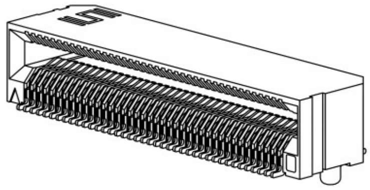


Figure 73: Right angle mini-card connector (Source: Samtec)

# 8. Control registers and EEPROM

## 8.1. Memory map

The epc660 control registers (RAM) are used for controlling all features of the chip. They are organized as 256x8 bit into 0x00 ... 0xFF address locations. The address space 0x80 ... 0xFF is EEPROM backed-up. Parameters in this section can be stored permanently between the power off/on cycles. All registers can be accessed through I²C interface by the application CPU (see chapter 5.4. I2C Slave). Multiple byte registers are stored as big-endian in the memory map.

It is further split into two pages as Control Page (0x00 ... 0x7F) and as EEPROM Page-0 (0x80 ... 0xFF).

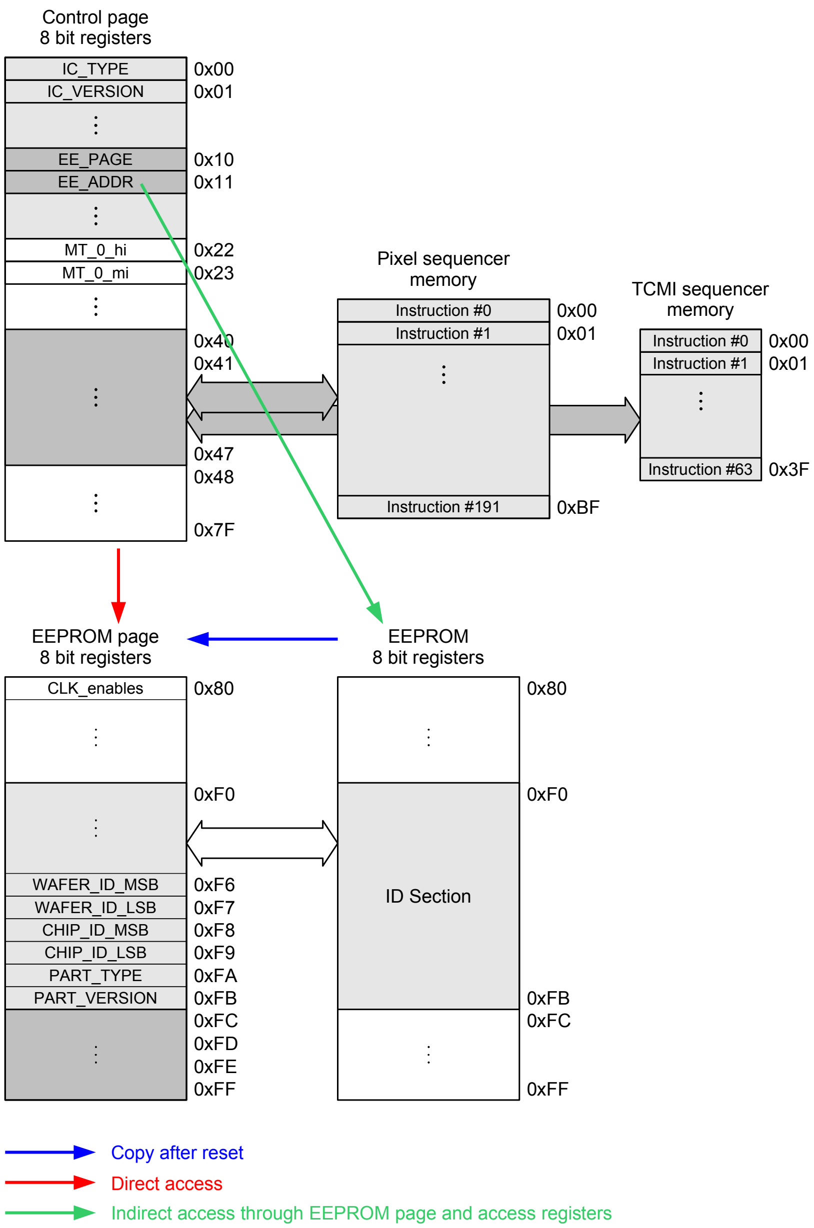



Figure 74: Memory Map

## 8.2. Control page

The control page contains only RW accessible configuration registers with default values during startup. The content can be changed via I²C interface. The changed values are preserved as long as the IC is powered. The registers reset back to their default values with a power-reset cycle or with a reset cycle alone.

## 8.3. EEPROM page, EEPROM and its charge pump

The epc660 has an 128x8-bit embedded EEPROM to store operation parameters, trimming and calibration values.

The EEPROM page address space (RAM) contains RW accessible configuration registers. Their content can be saved into the on-chip 128 x 8-bit EEPROM. It is mapped exactly one-to-one in the same RAM address space. There are various methods how the content of EEPROM can be saved or copied back and forth with the actual RW configuration registers (for details, refer to chapter 5.4.7., Control commands).

Each time a single or multiple I²C byte-write command sequence is applied to the EEPROM address space, a charge-pump for program and erase operations is automatically enabled before the EEPROM write access starts. It is disabled after the write access is finished.

The application software must respect a 20ms/byte write time (program/erase cycle) when altering the EEPROM content through I²C interface. When i2c_clock_stretch_en = 1, the I²C slave interface will stretch the SCK line while the EEPROM write continues. The application CPU can monitor the SCK line with its I²C master interface before sending the next byte-write command. If the application CPU does not support the I²C clock stretching feature, the application should first set i2c_clock_stretch_en = 0 and keep a timer for waiting 20ms between every I²C byte-write command sequence.

**IMPORTANT:**

- The built-in charge-pump for the EEPROM is a noisy circuit which shall not be enabled during frame acquisition. It can degrade the performance of the measurements. Therefore, the application must not alter the content of the EEPROM while a frame acquisition is going on.
- The number of WRITE cycles into the EEPROM is limited. It should not exceed 100 WRITE operations.

### 8.4. Register map

| Address | Register name | Ref. | Description | Access standard | Access advanced |
|---|---|---|---|---|---|
| **Memory and EEPROM controller registers** | | | | | |
| 0x00 | IC_TYPE | 8.6.1. | Unique IC Type for device family identification | R | R |
| 0x01 | IC_VERSION | 8.6.1. | Unique IC Version for device mask identification | R | R |
| 0x02<br>…<br>0x10 | reserved | | | --- | --- |
| 0x11 | EE_ADDR | 8.6.2. | Address for indirect read/write access to EEPROM | RW | RW |
| 0x12 | EE_DATA | 8.6.2. | Data for indirect read/write access to EEPROM | RW | RW |
| 0x13 | EE_MASK | 8.6.2. | Mask bits for indirect write access to EEPROM | RW | RW |
| 0x14 | PROG_EE_REQ | 8.6.3. | Requests programming of segments of the EEPROM | RW | RW |
| 0x15<br>…<br>0x1F | reserved | | | --- | --- |
| **Strap registers** | | | | | |
| 0x20<br>0x21 | Strap_hi<br>Strap_lo | 8.6.4. | Strap Scan Value Hold Register | R | R |
| **Modulation table registers** | | | | | |
| 0x22<br>0x23<br>0x24 | MT_0_hi<br>reserved<br>MT_0_lo | 8.6.5. | MT-vector: Sine/PN mode DCS0 | RW | RW |
| 0x25<br>0x26<br>0x27 | MT_1_hi<br>reserved<br>MT_1_lo | 8.6.5. | MT-vector: Sine/PN mode DCS1 | RW | RW |
| 0x28<br>0x29<br>0x2A | MT_2_hi<br>reserved<br>MT_2_lo | 8.6.5. | MT-vector: Sine/PN mode DCS2 | RW | RW |
| 0x2B<br>0x2C<br>0x2D | MT_3_hi<br>reserved<br>MT_3_lo | 8.6.5. | MT-vector: Sine/PN mode DCS3 | RW | RW |
| 0x2E<br>…<br>0x5F | reserved | | | --- | --- |
| **Temperature sensor registers** | | | | | |
| 0x60<br>0x61 | Sum_Temp_tl_hi<br>Sum_Temp_tl_lo | 8.6.6. | Temperature sensor top-left | R | R |
| 0x62<br>0x63 | Sum_Temp_tr_hi<br>Sum_Temp_tr_lo | 8.6.6. | Temperature sensor top-right | R | R |
| 0x64<br>0x65 | Sum_Temp_bl_hi<br>Sum_Temp_bl_lo | 8.6.6. | Temperature sensor bottom-left | R | R |
| 0x66<br>0x67 | Sum_Temp_br_hi<br>Sum_Temp_br_lo | 8.6.6. | Temperature sensor bottom-right | R | R |
| 0x68<br>...<br>0x6F | reserved | | | --- | --- |

Table 41: Address map of the control page (0x00 ~ 0x7F)

| Address | Register name | Ref. | Description | Access standard | Access advanced |
|---|---|---|---|---|---|
| **DLL manual control registers** | | | | | |
| 0x70 | DLL_status | 8.6.7. | DLL manual control: Status | --- | R |
| 0x71<br>0x72 | DLL_fine_ctrl_ext_hi<br>DLL_fine_ctrl_ext_lo | 8.6.8. | DLL manual control: External fine control | --- | RW |
| 0x73 | DLL_coarse_ctrl_ext | 8.6.9. | DLL manual control: External coarse control | --- | RW |
| 0x74<br>0x75 | DLL_fine_low_bank_rb_hi<br>DLL_fine_low_bank_rb_lo | 8.6.10. | DLL manual control: Fine lowest bank read back | --- | R |
| 0x76<br>0x77 | DLL_fine_bank_rb_hi<br>DLL_fine_bank_rb_lo | 8.6.10. | DLL manual control: Fine bank read back | --- | R |
| 0x78 | DLL_fine_page_rb | 8.6.10. | DLL manual control: Fine section page read back | --- | R |
| 0x79 | DLL_coarse_rb | 8.6.11. | DLL manual control: Coarse control read back | --- | R |
| 0x7A<br>0x7C | reserved | | | --- | --- |
| **Configuration mode registers** | | | | | |
| 0x7D | CFG_Mode_Control | 8.6.12. | Configuration mode: Control register | --- | RW |
| 0x7E<br>0x7F | reserved | | | --- | --- |

Table 41 cont.: Address map of the control page (0x00 ~ 0x7F)

| Address | Register name | Ref. | Description | Access standard | Access advanced |
|---|---|---|---|---|---|
| **CGU registers, SEG0 *** | | | | | |
| 0x80 | CLK_enables | 8.7.1. | CGU register, SEG0.*: Clock enable | RW | RW |
| 0x81<br>…<br>0x84 | reserved | | | --- | --- |
| 0x85 | MOD_CLK_divider | 8.7.2. | CGU register, SEG0.*: MOD clock divider | RW | RW |
| 0x86<br>…<br>0x88 | reserved | | | --- | --- |
| 0x89 | TCMI_CLK_divider | 8.7.3. | CGU register, SEG0.*: TCMI clock | RW | RW |
| 0x8A | reserved | | | --- | --- |
| **Modulator/demodulator registers, SEG1 *** | | | | | |
| 0x8B | Demodulation_delays | 8.7.4. | Mod./demod. registers, SEG1 *: Delays | RW | RW |
| 0x8C<br>0x8D | PN_POLY_hi<br>PN_POLY_lo | 8.7.5. | Mod./demod. registers, SEG1 *: PN polynomial | RW | RW |
| 0x8E<br>0x8F | reserved | | | --- | --- |
| 0x90 | LED_driver | 8.7.6. | Mod./demod. registers, SEG1 *: LED driver | RW | RW |
| 0x92 | MOD_Control ** | 8.7.7. | Mod./demod. registers, SEG1 *: Modulation select | RW | RW |
| 0x93 | Dist_offset ** | 8.7.8. | Mod. /demod. registers, SEG1 *: Distance offset | RW | RW |
| **Pixel operating and readout control registers** | | | | | |
| 0x94 | Resolution_reduction ** | 8.7.9. | Pixel operating and readout control | RW | RW |
| **Frame column / row mapping registers** | | | | | |
| 0x95 | Readout_dir ** | 8.7.10. | Frame column / row mapping: readout direction | RW | RW |
| 0x96<br>0x97 | ROI_tl_x_hi **<br>ROI_tl_x_lo ** | 8.7.11. | Frame column / row mapping: ROI top-left x | RW | RW |
| 0x98<br>0x99 | ROI_br_x_hi **<br>ROI_br_x_lo ** | 8.7.11. | Frame column / row mapping: ROI bottom-right x | RW | RW |
| 0x9A | ROI_tl_y ** | 8.7.11. | Frame column / row mapping: ROI top-left y | RW | RW |
| 0x9B | ROI_br_y ** | 8.7.11. | Frame column / row mapping: ROI bottom-right y | RW | RW |
| **Shutter control registers** | | | | | |
| 0x9C<br>0x9D | reserved | | | --- | --- |
| 0x9E<br>0x9F | Int_len_mgx1_hi **<br>Int_len_mgx1_lo ** | 8.7.12. | Shutter control: Integration time mgx1 | RW | RW |
| 0xA0<br>0xA1 | INTM_hi **<br>INTM_lo ** | 8.7.13. | Shutter control: Integration time multiplier | RW | RW |
| 0xA2<br>0xA3 | Int_len_hi **<br>Int_len_lo ** | 8.7.14. | Shutter control: Integration time | RW | RW |
| 0xA4 | Shutter_Control | 8.7.15. | Shutter control: Video mode / SW control | RW | RW |
| **Power control registers** | | | | | |
| 0xA5 | Power_Control (analog) | 8.7.16. | Power control | RW | RW |
| **DLL synchronization registers** | | | | | |
| 0xA6<br>0xA7 | DLL_en_del_hi<br>DLL_en_del_lo | 8.7.17. | DLL synchronization: DLL pre-synchronization time | RW | RW |
| 0xA8<br>0xA9 | DLL_en_hi<br>DLL_en_lo | 8.7.17. | DLL synchronization: DLL synchronization time | RW | RW |
| 0xAA<br>0xAB | DLL_measurement_rate_hi **<br>DLL_measurement_rate_lo ** | 8.7.18. | DLL synchronization: DLL measurement rate | RW | RW |

Table 42: Address map of EEPROM page-0 (0x80 ~ 0xFF)

| Address | Register name | Ref. | Description | Access standard | Access advanced |
|---|---|---|---|---|---|
| **DLL control registers** | | | | | |
| 0xAC | reserved | | | --- | --- |
| 0xAD | DLL_match_width | 8.7.19. | DLL control: Match width | --- | RW |
| 0xAE | DLL_control | 8.7.20. | DLL control | RW | RW |
| 0xAF<br>...<br>0xC9 | reserved | | | --- | --- |
| **I²C registers, SEG4 *** | | | | | |
| 0xCA | I²C_address | 8.7.21. | I²C register, SEG4 *: Address | R | R |
| 0xCB | I²C_control | 8.7.21. | I²C register, SEG4 *: Control | RW | RW |
| **TCMI registers** | | | | | |
| 0xCC | TCMI_polarity | 8.7.22. | TCMI register | RW | RW |
| 0xCD<br>…<br>0xE7 | reserved | | | --- | --- |
| **Temperature sensor calibration registers (factory settings)** | | | | | |
| 0xE8 | Temp_tl_cal1 | 8.7.23. | Temperature sensor top-left calibration point 1 | R | RW |
| 0xE9 | Temp_tl_cal2 | 8.7.23. | Temperature sensor top-left calibration point 2 | R | RW |
| 0xEA | Temp_tr_cal1 | 8.7.23. | Temperature sensor top-right calibration point 1 | R | RW |
| 0xEB | Temp_tr_cal2 | 8.7.23. | Temperature sensor top-right calibration point 2 | R | RW |
| 0xEC | Temp_bl_cal1 | 8.7.23. | Temperature sensor bottom-left calibration point 1 | R | RW |
| 0xED | Temp_bl_cal2 | 8.7.23. | Temperature sensor bottom-left calibration point 2 | R | RW |
| 0xEE | Temp_br_cal1 | 8.7.23. | Temperature sensor bottom-right calibration point 1 | R | RW |
| 0xEF | Temp_br_cal2 | 8.7.23. | Temperature sensor bottom-right calibration point 2 | R | RW |
| **Free EEPROM registers (for customer use), SEG6 *** | | | | | |
| 0xF0 | User_1 | 8.7.24. | User register 1 for user data | RW | RW |
| 0xF1 | User_2 | 8.7.24. | User register 2 for user data | RW | RW |
| **CHIP ID registers (factory settings), SEG7 *** | | | | | |
| 0xF2<br>...<br>0xF4 | reserved | | | --- | --- |
| 0xF5 | ENGINEERING_ID | 8.7.25. | Chip ID register SEG6 *: Engineering ID for MPWs | R | R |
| 0xF6<br>0xF7 | WAFER_ID_MSB<br>WAFER_ID_LSB | 8.7.26. | Chip ID register SEG6 *: Wafer ID | R | R |
| 0xF8<br>0xF9 | CHIP_ID_MSB<br>CHIP_ID_LSB | 8.7.27. | Chip ID register SEG6 *: Chip ID on wafer | R | R |
| 0xFA | PART_TYPE | 8.7.28. | Chip ID register SEG6 *: Part type | R | R |
| 0xFB | PART_VERSION | 8.7.28. | Chip ID register SEG6 *: Part version | R | R |
| 0xFC<br>…<br>0xFF | reserved | | | --- | --- |

Table 42 cont.: Address map of EEPROM page-0 (0x80 ~ 0xFF)

Notes:

* EEPROM protected segments. Protection will be enabled during production testing.

** Shadowed (double-buffered) registers which can be updated on-the-fly via I²C, while a frame acquisition is going on.
The new values get copied and used at the start of the next frame.
Shadow registers can be bypassed by setting Pixel_test(0x7B).shadow_regs_bypas s = 1. In this case, written values take effect immediately.

### 8.5. Register description: Explanation example

| **Register name: MOD_Control **** | | | | **Description: Mod./demod. registers, SEG1 *: Modulation select** | | | | **Address: 0x92** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | mod_sel | | dcs_sel | | int_len _mgx1_en | lfsr_sel | | |
| Operation | Access option for the register<br>■ R Read<br>■ W Write<br>■ RW Read or Write<br>■ (RW) Note: Needs epc support to access this function by user: Read or Write | | | | | | | |
| Default | Default factory setting of the register or register bits | | | | | | | |

### 8.6. Register description: Control Page (0x00 ~ 0x7F)

#### 8.6.1. IC_

| **IC_TYPE** | **Unique IC Type for device family identification** | **0x00** |
|---|---|---|
| Operation | R | |
| Default | according to type | |

| **IC_VERSION** | **Unique IC Version for device mask identification** | **0x01** |
|---|---|---|
| Operation | R | |
| Default | according to mask | |

#### 8.6.2. EE_ (ADDR, DATA, MASK)

| **EE_ADDR** | **Address for indirect read/write access to EEPROM** | **0x11** |
|---|---|---|
| Operation | RW | |
| Default | 0x00 (note 1) | |

Note1: Default (reset) value is internally modified during start-up sequence. Therefore, when this register is read via $I^2C$, a different value may be observed.

| **EE_DATA** | **Data for indirect read/write access to EEPROM** | **0x12** |
|---|---|---|
| Operation | RW | |
| Default | 0x3F | |

| **EE_MASK** | **Mask bits for indirect write access to EEPROM** | **0x13** |
|---|---|---|
| Operation | RW | |
| Default | 0x00 | |

#### 8.6.3. PROG_EE_REQ

| **PROG_EE_REQ** | **Requests programming of segments of the EEPROM** | **0x14** |
|---|---|---|
| Operation | RW | |
| Default | 0x00 | |

#### 8.6.4. Strap_

| Strap_hi | | | | Strap Scan Value Hold Register (high) | | | | 0x20 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | strap14 | strap13 | reserved | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see below Strap_lo register

| Strap_lo | | | | Strap Scan Value Hold Register (low) | | | | 0x21 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | | | | strap0 |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

| | |
|---|---|
| Strap15: | reserved, displays pin XYSNC_SAT_CFG |
| Strap14: | I²C Device Address: A1, on HYSNC_A1 |
| Strap13: | I²C Device Address: A0, on VYSNC_PSDIAG_A0 |
| Strap12: | reserved, displays pin DCLK |
| Strap11 ... Strap0: | reserved, displays pins DATA[11:0] |

Note 1: Default start-up values of these registers are only valid until end of reset phase. Values might be overwritten by external pull-up resistors during strap scan phase when reset is released.

Note 2: Read access on this register does not initiate a strap scan phase. This register simply holds the last scanned value when reset is released.

#### 8.6.5. MT_

| MT_0_hi | | | | MT-vector: Sine/PN mode DCS 0 (high) | | | | 0x22 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit name | reserved | | readout_mode | | mgx1_dcs_num | | mgx0_dcs_num | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 0x34 | | | | | | | |

Modulation Table Vector bit definitions:

readout_mode: Readout mode of the vector
00: reserved
01: single-ended readout from MGA
10: single-ended readout from MGB
11: differential-ended readout from MGA and MGB

mgx1_dcs_num: DCS frame number of the vector for mgx1 modulator outputs (mga1, mgb1)
00: DCS 0
01: DCS 1
10: DCS 2
11: DCS 3

mgx0_dcs_num: DCS frame number of the vector for mgx0 modulator outputs (mga0, mgb0)
00: DCS 0
01: DCS 1
10: DCS 2
11: DCS 3

| MT_0_lo | | | | MT-vector: Sine/PN mode DCS 0 (low) | | | | 0x24 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | override_led_ modulated_ B&W | reserved | override_led_ off_integ/ override_led_ on_mga_b&w | reserved | override_mgb _on_integ | override_mga _on_integ | override_mgb _off | override_mga _off |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

override_led_modulated_B&W:
- Sine/PN/B&W: When LED_driver.led_on = 1 (torch function):
  0: LED/LD on (DC) during integration period
  1: LED/LD is modulated during integration period

override_led_off_integ/override_led_on_mga_b&w:
- Sine/PN/B&W:
  0: LED/LD is modulated
  1: LED/LD stays off entire integration time

override_mgb_on_integ: MGB stays on (conducting) entire integration time

override_mga_on_integ: MGA stays on (conducting) entire integration time

override_mgb_off: MGB stays off (non-conducting) all the time

override_mga_off: MGA stays off (non-conducting) all the time

Note 1: Priorities: b0 over b2, b1 over b3.

Note 2: b5..b0 are ignored during DLL synchronization

| MT_1_hi | | | | MT-vector: Sine/PN mode DCS 1 (high) | | | | 0x25 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit name | reserved | | readout_mode | | mgx1_dcs_num | | mgx0_dcs_num | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| | 0x31 | | | | | | | |

Note: Description see MT_0_hi

| MT_1_lo | | | | MT-vector: Sine/PN mode DCS 1 (low) | | | | 0x27 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | override_led_ modulated_ B&W | reserved | override_led_ off_integ/ override_led_ on_mga_b&w | reserved | override_mgb _on_integ | override_mga _on_integ | override_mgb _off | override_mga _off |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see MT_0_lo

| MT_2_hi | | | | MT-vector: Sine/PN mode DCS 2 (high) | | | | 0x28 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit name | reserved | | readout_mode | | mgx1_dcs_num | | mgx0_dcs_num | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| | 0x3E | | | | | | | |

Note: Description see MT_0_hi

| MT_2_lo | | | | MT-vector: Sine/PN mode DCS 2 (low) | | | | 0x2A |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | override_led_ modulated_ B&W | reserved | override_led_ off_integ/ override_led_ on_mga_b&w | reserved | override_mgb _on_integ | override_mga _on_integ | override_mgb _off | override_mga _off |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see MT_0_lo

| MT_3_hi | | | | MT-vector: Sine/PN mode DCS 3 (high) | | | | 0x2B |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit name | reserved | | readout_mode | | mgx1_dcs_num | | mgx0_dcs_num | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| | 0x3B | | | | | | | |

Note: Description see MT_0_hi

| MT_3_lo | | | | MT-vector: Sine/PN mode DCS 3 (low) | | | | 0x2D |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | override_led_ modulated_ B&W | reserved | override_led_ off_integ/ override_led_ on_mga_b&w | reserved | override_mgb _on_integ | override_mga _on_integ | override_mgb _off | override_mga _off |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see MT_0_lo

#### 8.6.6. Sum_Temp_

| Sum_Temp_tl_hi | | | | Temperature sensor top-left (high) | | | | 0x60 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | Sum_Temp_tl[13:8] | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see Sum_Temp_tl_lo

| Sum_Temp_tl_lo | | | | Temperature sensor top-left (low) | | | | 0x61 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | Sum_Temp_tl[7:0] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Sum_Temp_tl_lo: Sum of 6 values (every 4th reading) of temperature sensor top-left

| Sum_Temp_tr_hi | | | | Temperature sensor top-right (high) | | | | 0x62 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | Sum_Temp_tr[13:8] | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see Sum_Temp_tr_lo

| Sum_Temp_tr_lo | | | | Temperature sensor top-right (low) | | | | 0x63 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | Sum_Temp_tr[7:0] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Sum_Temp_top-right: Sum of 6 values (every 4th reading) of temperature sensor top-right

| Sum_Temp_bl_hi | | | | Temperature sensor bottom-left (high) | | | | 0x64 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | Sum_Temp_bl[13:8] | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see Sum_Temp_bl_lo

| Sum_Temp_bl_lo | | | | Temperature sensor bottom-left (low) | | | | 0x65 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | Sum_Temp_bl[7:0] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Sum_Temp_bottom-left: Sum of 6 values (every 4th reading) of temperature sensor bottom-left

| Sum_Temp_br_hi | | | | Temperature sensor bottom-right (high) | | | | 0x66 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | Sum_Temp_br[13:8] | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see Sum_Temp_br_lo

| Sum_Temp_br_lo | | | | Temperature sensor bottom-right (low) | | | | 0x67 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | Sum_Temp_br[7:0] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Sum_Temp_bottom-right: Sum of 6 values (every 4th reading) of temperature sensor bottom-right

### 8.6.7. DLL_status

| **DLL_status** | | | | **DLL manual control: Status** | | | | **0x70** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | dll_fb_lead_stat | dll_ref_lead_stat | dll_error_stat | dll_filter_stat | dll_match_stat | dll_lock_stat |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

dll_fb_lead_stat: Live status signal indicating the feedback waveform is leading over the reference. Used to implement the delay control by SW.

dll_ref_lead_stat: Live status signal indicating the reference waveform is leading over the feedback. Used to implement the delay control by SW.

dll_error_stat: Asserted when the delay to be compensated is out of reach for the delay line.

dll_filter_stat: Asserted when the DLL jitter filter is active.

dll_match_stat: Asserted when the residual delay between the reference and feedback is less than ~120ps.

dll_lock_stat Asserted when the residual delay between the reference and feedback is less than ~2.4ns.

### 8.6.8. DLL_fine_ctrl_ext

| **DLL_fine_ctrl_ext_hi** | | | | **DLL manual control: External fine control (high)** | | | | **0x71** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | | | | | DLL_fine_ctrl_ext[9:8] | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see DLL_fine_ctrl_ext_lo

| **DLL_fine_ctrl_ext_lo** | | | | **DLL manual control: External fine control (low)** | | | | **0x72** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | DLL_fine_ctrl_ext[7:0] | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

DLL_fine_ctrl_ext: Fine control from user. Valid only when DLL_control.dll_ctrl_ext = 1 (see section 8.7.20., DLL_control)

### 8.6.9. DLL_coarse_ctrl_ext

| **DLL_coarse_ctrl_ext** | | | | **DLL manual control: External coarse control** | | | | **0x73** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | dll_coarse_ctrl_ext | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

dll_coarse_ctrl_ext: Coarse control from user. Valid only when DLL_control.dll_ctrl_ext = 1 (see section 8.7.20., DLL_control)

### 8.6.10. DLL_fine_

| DLL_fine_low_bank_rb_hi | | | | DLL manual control: Fine lowest bank read back (high) | | | | 0x74 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | DLL_fine_low_bank_rb[15:8] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0x00 | | | | | | | |

| DLL_fine_low_bank_rb_lo | | | | DLL manual control: Fine lowest bank read back (low) | | | | 0x75 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | DLL_fine_low_bank_rb[7:0] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0x00 | | | | | | | |

| DLL_fine_bank_rb_hi | | | | DLL manual control: Fine bank read back (high) | | | | 0x76 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | DLL_fine_bank_rb[15:8] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0x00 | | | | | | | |

| DLL_fine_bank_rb_lo | | | | DLL manual control: Fine bank read back (low) | | | | 0x77 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | DLL_fine_bank_rb[7:0] | | | | | | | |
| Operation | R | | | | | | | |
| Default | 0x00 | | | | | | | |

| DLL_fine_page_rb | | | | DLL manual control: Fine section page read back | | | | 0x78 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | | | dll_fine_page_rb | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | x | x |
| | 0x00 | | | | | | | |

dll_fine_page_rb: Fine section page read back

### 8.6.11. DLL_coarse_rb

| DLL_coarse_rb | | | | DLL manual control: Coarse control read back | | | | 0x79 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | dll_coarse_rb[5:0] | | | | | |
| Operation | R | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

#### 8.6.12. CFG_Mode_

| CFG_Mode_Control | | | | Configuration mode: Control register | | | | 0x7D |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | sys_clk _bypass | pll_bypass | pll_en | reserved | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0x04 (Note 1) | | | | | | | |

sys_clk_bypass: Bypass OSC clock directly to the sys_clk output
0: normal
1: bypass (sys_clk = osc_clk)

pll_bypass: Bypass OSC clock directly to the PLL output as clock source to the CGU
0: normal
1: bypass (pll_clk = osc_clk)

pll_en: Enable PLL block
0: disable
1: enable

Note 1: Default (reset) value is internally modified during start-up sequence. Therefore, when this register is read via I²C, a different value will be observed.

### 8.7. Register description: EEPROM Page-0 (0x80 ~ 0xFF)

#### 8.7.1. CLK_enables

| CLK_enables | | | | CGU register, SEG0.*: Clock enable | | | | 0x80 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | all_clks_en | mod_clk_sel | reseved | | | | mod_clk_en | dll_clk_en |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0x3F | | | | | | | |

CGU clock enable register:

all_clks_en: Force all clock enables
0: disable
1: enable

mod_clk_sel: Select modulator clock source
0: mod_clk, from CGU
1: ext_mod_clk, from MODCLK input pin

Note: Note: sys_clk is always running. It cannot be disabled.

#### 8.7.2. MOD_CLK_divider

| MOD_CLK_divider | | | | CGU register, SEG0.*: MOD clock divider | | | | 0x85 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | mod_clk_div | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

mod_clk_div: Provides the clock to the LED / pixel field modulator/demodulator circuits, respectively, by integer division of the PLL clock.
Default = 0: $f_{mod\_clk}$ = 80MHz / (mod_clk_div + 1) = 80MHz / 1 = 80MHz

### 8.7.3. TCMI_CLK_divider

| **TCMI_CLK_divider** | | | | **CGU register, SEG0.*: TCMI clock** | | | | **0x89** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | dclk_skew _en | tcmi_drv2 _en | reserved | tcmi_clk_div | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0x01 | | | | | | | |

dclk_skew_en: DCLK skew enable
0: disable
1: enable

Note1: Use dclk_skew_en = 1 to delay DCLK edge (typ. 2ns) to compensate PCB delays.
Might be particularly useful when tcmi_clk_div = 0 (divide by 1).
When set normal, DCLK edge is centered with respect to other TCMI *SYNC*, DATA[11:0] outputs.

tcmi_drv2_en: TCMI pins driver-2 enable
0: disable
1: enable

Note2: Use tcmi_drv2_en = 1 to enable driver-2 of all TCMI output pins. This may require special VDDIO/VSSIO supply coupling to avoid spikes - use with caution!

tcmi_clk_div: Provides the clock to the TCMI by integer division of the PLL clock.
Default = 1: $f_{tcmi\_clk}$ = 80MHz / (tcmi_clk_div + 1) = 80MHz / 2 = 40MHz

### 8.7.4. Demodulation_delays

| **Demodulation_delays** | | | | **Mod./demod. registers, SEG1 *: Delays** | | | | **0x8B** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | mgx_del_sel | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0x01 | | | | | | | |

mgx_del_sel Number of pll_clk period delays on the abd, abg, mga, mgb signals in order to compensate internal LED driver delay (all modulation modes)
0: no delay
1: 1 clock
2: 2 clocks
…
12: 12 clocks
13 … 15: reserved

Note 1: When DLL is enabled, mgx_del_sel must be initialized to a higher value with respect to the external LED/LD circuit's initial delay + max. delay deviation over the system's operating conditions. Minimum recommendation: 2.

Note 2: When mgx_del_sel set > 12 (i.e. 13, 14 or 15), delay is internally limited to 12 pll_clk periods.

### 8.7.5. PN_

| **PN_POLY_hi** | | | | **Mod./demod. registers, SEG1 *: PN polynomial (high)** | | | | **0x8C** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | pn_poly[14:8] | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | 0x08 | | | | | | | |

Note: Description see PN_POLY_lo

| PN_POLY_lo | | | | Mod./demod. registers, SEG1 *: PN polynomial (low) | | | | 0x8D |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | pn_poly[7:0] | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| | 0x05 | | | | | | | |

pn_poly: PN Polynomial configuration (default: 2'053)

**8.7.6. LED_driver**

| LED_driver | | | | Mod./demod. registers, SEG1 *: LED driver | | | | 0x90 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | led_drv_sel | | reserved | led_on | reserved | led_drv_en | led_inv_en | led_ssr_en |
| Operation | RW | | | | | | | |
| Default | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0xC4 | | | | | | | |

led_drv_sel: LED driver output max. current select e.g. for LED current 200mA ≙ 100%
00: 33%
01: 46%
10: 64%
11: 100%

led_on: LED/LD permanently on (DC) (torch function)
0: off
1: on

led_drv_en: LED driver internal power enable (independent of the modulator)
0: disable
1: enable

led_inv_en: LED signal inversion. Inversion of the output levels on pin LED.
0: disable (when active: signal on pin LED = 0, VSSLED)
1: enable (when active: active: signal on pin LED = 1, open-drain)

Note: When led_inv_en = 0: It can directly sink the LED turn-on current (up to 200mA peak) without additional external driver.
When led_inv_en = 1: It needs an external pull-up resistor (open-drain) to drive the inverted value for the external LED drive circuitry.

led_ssr_en: LED slow slew rate enable.
0: disable
1: enable

### 8.7.7. MOD_Control **

| **MOD_Control** ** | | | | **Mod./demod. registers, SEG1 *: Modulation select** | | | | **0x92** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | mod_sel | | dcs_sel | | int_len _mgx1_en | lfsr_sel | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 0x34 | | | | | | | |

mod_sel: Modulation select
00: Sinusoidal mode
01: PN mode
10: reserved
11: reserved

dcs_sel: Number of DCS read-outs select
00: 1x DCSx: DCS0
01: 2x DCSx: DCS0, DCS1
10: reserved
11: 4x DCSx: DCS0, DCS1, DCS2, DCS4

int_len_mgx1_en: Int_Len_mgx1_hi/lo enable (refer to DUAL MGX mode – single DCS acquisition with different integration time)
0: disable
1: enable

Note: int_len_mgx1_en must be enabled only when dual_mgx_mode = 1, otherwise it is not effective (see Figure 33)

lfsr_sel: Number of LFSR stages select
000: reserved
001: 14
010: 13
011: 12
100: 11
101: 10
110: 9
111: 8

### 8.7.8. Dist_offset **

| **Dist_offset** ** | | | | **Mod./demod. registers, SEG1 *: Distance offset** | | | | **0x93** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | dist_offset | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

dist_offset: Number of pll_clk period delays on the abd, abg, mga, mgb signals in order to capture reflected LED/LD light from faraway objects with PN mode

Note: It is effective in all modulation modes.
Therefore, it is recommended to set it to 0 for all modulation modes other than PN mode.

### 8.7.9. Resolution_reduction **

| **Resolution_reduction** ** | | | | **Pixel operating and readout control** | | | | **0x94** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | dual_mgx_mode | reserved | pixel_bin_mode | | res_red_row | | res_red_col | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

dual_mgx_mode: Dual MGX mode (refer to chapter Fehler: Referenz nicht gefunden, Fehler: Referenz nicht gefunden)
0: single
1: dual

pixel_bin_mode: Pixel binning mode
00: no binning
01: horizontal binning (x-axis) (only when res_red_col <> 00)
10: vertical binning (y-axis) (only when res_red_row <> 00)
11: horizontal + vertical binning (only when res_red_col <> 00 AND res_red_row <> 00)

Note 1: Pixel binning cannot be used with dual MGX modes.

res_red_row: Row reduction: Selects resolution on y-axis
00: full (0, 1, 2, …)
01: half (0, 2, 4, …)
10: quarter (0, 4, 8, …)
11: one eight (0, 8, 16, …)

res_red_col: Column reduction: Selects resolution on x-axis
00: full (0, 1, 2, …)
01: half (0, 2, 4, …)
10: reserved
11: reserved

Note 2: If binning is selected, then the corresponding reduction mode should also be chosen (res_red_row, res_red_col).

### 8.7.10. Readout_dir **

| **Readout_dir** ** | | | | **Frame column / row mapping: readout direction** | | | | **0x95** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | | | col_dir_sel | reserved |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0x03 | | | | | | | |

col_dir_sel: Column readout direction
0: decrement
1: increment

### 8.7.11. ROI_

| **ROI_tl_x_hi** ** | | | | **Frame column / row mapping: ROI top-left x (high)** | | | | **0x96** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | | | | | | ROI_tl_x[8] |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see ROI_tl_x_lo **

| ROI_tl_x_lo ** | | | | Frame column / row mapping: ROI top-left x (low) | | | | 0x97 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | ROI_tl_x[7:0] | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0x04 (default: 4) | | | | | | | |

ROI_tl_x: Refer to chapter 5.6.3., section ROI (Region of Interest)

| ROI_br_x_hi ** | | | | Frame column / row mapping: ROI bottom-right x (high) | | | | 0x98 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | | | | | | ROI_br_x[8] |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0x01 | | | | | | | |

Note: Description see ROI_br_x_lo **

| ROI_br_x_lo ** | | | | Frame column / row mapping: ROI bottom-right x (low) | | | | 0x99 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | ROI_br_x[7:0] | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0x43 (default: 323) | | | | | | | |

ROI_br_x: Refer to chapter 5.6.3., section ROI (Region of Interest)

| ROI_tl_y ** | | | | Frame column / row mapping: ROI top-left y | | | | 0x9A |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | ROI_tl_y | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| | 0x06 (default: 6) | | | | | | | |

ROI_tl_y: Refer to chapter 5.6.3., section ROI (Region of Interest)

| ROI_br_y ** | | | | Frame column / row mapping: ROI bottom-right y | | | | 0x9B |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | ROI_br_y | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| | 0x7D (default: 125) | | | | | | | |

ROI_br_y: Refer to chapter 5.6.3., section ROI (Region of Interest)

#### 8.7.12. Int_len_mgx1_

| Int_len_mgx1_hi ** | | | | Shutter control: Integration time mgx1 (high) | | | | 0x9E |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 0x07 | | | | | | | |

Note: Description see Int_len_mgx1_lo **

| Int_len_mgx1_lo ** | | | | Shutter control: Integration time mgx1 (low) | | | | 0x9F |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Operation | RW | | | | | | | |
| Default | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 0xFF | | | | | | | |

Int_len_mgx1: Number of mod_clk periods for the integration time of mgx1 (Default: 2'047).
See Int_len_hi/lo registers (0xA2/0xA3) for functional definition details.

Note: int_len_mgx1_en = 0: These registers are updated simultaneously to the same value every time the Int_len_hi/lo (0xA2/0xA3) registers are written by $I^2C$.

int_len_mgx1_en =1: Int_len_mgx1_hi/lo can be updated independent of Int_len_1_hi/lo. This enables asymmetric integration times for upper and lower pixels rows

#### 8.7.13. INTM_

| INTM_hi ** | | | | Shutter control: Integration time multiplier (high) | | | | 0xA0 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit name | reserved | | | | | | INTM[9:8] | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see INTM_lo **

| INTM_lo ** | | | | Shutter control: Integration time multiplier (low) | | | | 0xA1 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | INTM[7:0] | | | | | | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0x01 | | | | | | | |

INTM: Multiplier of the basic integration length set in Int_len

Note: Minimum INTM value = 1

#### 8.7.14. Int_len_

| Int_len_hi ** | | | | Shutter control: Integration time (high) | | | | 0xA2 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 0x07 | | | | | | | |

Note: Description see Int_len_lo **

| Int_len_lo ** | | | | Shutter control: Integration time(low) | | | | 0xA3 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Operation | RW | | | | | | | |
| Default | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 0xFF | | | | | | | |

Int_len: Number of mod_clk periods for the PN integration time.
Default = 2'047: Integration time = INTM * (Int_len + 1) * $t_{mod\_clk}$ = 25.6us

Note: Sine mode: (Int_len + 1) should be evenly divisible by 4.
Min Int_len = 7 (100ns).

PN mode (Int_len + 1) should be evenly divisible by 4 and correspond to the LSFR setting.
Min Int_len = 1'019 (12.75µs).

**8.7.15. Shutter_Control**

| Shutter_Control | | | | Shutter control: Video mode / SW control | | | | 0xA4 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | | | multi_frame_en | shutter_en |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

multi_frame_en: Multiple frames (video mode). Refer to chapter Fehler: Referenz nicht gefunden, Fehler: Referenz nicht gefunden.
0: disable
1: enable

shutter_en: Shutter release (SW control)
0: disable
1: enable: starts acquisition (auto cleared)

Note: shutter_en is not auto-cleared when multi_frame_en = 1

**8.7.16. Power_Control (analog)**

| Power_Control (analog) | | | | Power control | | | | 0xA5 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | reserved | See note below | | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 0x07 | | | | | | | |

Note: Refer to chapter 7.7., Power saving options, Table 39

**8.7.17. DLL_en_**

| DLL_en_del_hi | | | | DLL synchronization: DLL pre-synchronization (high) | | | | 0xA6 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0x03 | | | | | | | |

Note: Description see DLL_en_del_lo

| DLL_en_del_lo | | | | DLL synchronization: DLL pre-synchronization (low) | | | | 0xA7 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| | 0x1F | | | | | | | |

DLL_en_del: Number of mod_clk periods before the DLL enable (DLL pre-synchronization time).
Default = 799: delay = (DLL_en_del + 1) * $t_{mod\_clk}$ = 10us

| DLL_en_hi | | | | DLL synchronization: DLL synchronization time (high) | | | | 0xA8 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| | 0x09 | | | | | | | |

Note: Description see DLL_en_lo

| DLL_en_lo | | | | DLL synchronization: DLL synchronization time (low) | | | | 0xA9 |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Operation | RW | | | | | | | |
| Default | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| | 0x5F | | | | | | | |

DLL_en: Number mod_clk periods for the DLL synchronization time.
Default = 2'399: delay = (DLL_en + 1) * $t_{mod\_clk}$ = 30us

**8.7.18. DLL_measurement_rate_**

| DLL_measurement_rate_hi ** | | | | DLL synchronization: DLL measurement rate (high) | | | | 0xAA |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

Note: Description see DLL_measurement_rate_lo **

| DLL_measurement_rate_lo ** | | | | DLL synchronization: DLL measurement rate (low) | | | | 0xAB |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0x04 | | | | | | | |

DLL_measurement_rate: Number of DCS acquisitions for the DLL measurement to compensate for the phase shift of the LED driver path. When equal to 0, the modulator does not trigger the DLL to re-measure, it simply keeps the previously measured/programmed delay value during the following frame acquisition.

Note: The DLL measurement rate should be set to a multiple of the dcs frame number e.g. 4x DCS: DLL_measurement_rate = 4.

#### 8.7.19. DLL_match_width

| **DLL_match_width** | | | | **DLL control: Match width** | | | | **0xAD** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Operation | RW | | | | | | | |
| Default | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0xFF | | | | | | | |

#### 8.7.20. DLL_control

| **DLL_control** | | | | **DLL control** | | | | **0xAE** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | dll_ctrl _ext_en | dll_rate _cnt_rst | dll_ctrl _ext | dll_fb_pol | dll_bypass |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0x00 | | | | | | | |

DLL Control:

dll_ctrl_ext_en: DLL, LED driver, feedback amplifier enable during manual control for delay locking operation (i.e. Modulator control is completely bypassed)
0: disable
1: enable

dll_rate_cnt_rst: DLL measurement rate counter reset
0: normal
1: reset (auto cleared)

dll_ctrl_ext: Manual control of the delay line using DLL_*ctrl_ext* registers
0: normal
1: manual

dll_fb_pol: Inverts the polarity of signal at LEDFB pin
0: normal
1: inverted

Note 1: dll_fb_pol must be set with respect to LED_driver.led_inv_en value and the external LED/LD circuit feedback point connected to the LEDFB input (external LED/LD circuit can be inverting or non-inverting depending on the application).

dll_bypass: Bypass control of DLL for system delay compensation
0: normal
1. bypass

Note 2: When DLL is operated manually, application SW must set dll_ctrl_ext_en = 0 while keeping dll_ctrl_ext = 1 to freeze the locked value before starting frame acquisition.

#### 8.7.21. I²C_

| **I²C_address** | | | | **I²C register, SEG4 *: Address** | | | | **0xCA** |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | i2c_dev_adr(A6..A2) | | | | | reserved i2c_dev_adr(A1..A0) | |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 0x20 | | | | | | | |

i2c_dev_adr(A6..A2): I²C device address A6 … A2 of 7-bit I²C device address.

i2c_dev_adr(A1..A0): I²C device address A1, A0 of 7-bit I²C device address.
Programmable only during reset via strap pins using external pull-up resistors.

| I²C_control | | | | I²C register, SEG4 *: Control | | | | 0xCB |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | reserved | | | | | | i2c_spike _filter_en | i2c_clock _stretch_en |
| Operation | RW | | | | | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0x03 | | | | | | | |

i2c_spike_filter_en: I²C pins input spike filter
0: disabled (> 1MHz)
1: enabled (≤ 1MHz, FM+)

i2c_clock_stretch_en: I²C clock stretching
0: disabled
1: enabled

Note 1: When i2c_spike_filter_en = 0, SDA and SCL pins can be used up to 10MHz as inputs (driven rail-to-rail by a real CMOS driver, no pull-up). They can be used up to 2MHz as outputs.

#### 8.7.22. TCMI_polarity

| TCMI_polarity | | | | TCMI register | | | | 0xCC |
|---|---|---|---|---|---|---|---|---|
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | tcmi_data _sat_en | tcmi_xsync _sat_sel | reserved | tcmi_data _pol | tcmi_xsync _pol | tcmi_vsync _pol | tcmi_hsync _pol | tcmi_edge _sel |
| Operation | RW | | | | | | | |
| Default | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0x41 | | | | | | | |

tcmi_data_sat_en: Force DATA[11:0] = 0xFFF (unsigned) / 0x7FF (signed) during data-out operation when corresponding pixel is saturated
0: disabled
1: enabled

tcmi_xsync_sat_sel: Select XSYNC / SAT output pin function (SAT function is always active high)
0: XSYNC
1: SAT

tcmi_data_pol: DATA[11:0] unsigned/signed control (for signed output, MSB bit is inverted)
0: unsigned (0 ... 4'095)
1: signed (-2'048 ... 2'047)

tcmi_xsync_pol: XSYNC polarity
0: XSYNC active low
1: XSYNC active high

Note: tcmi_xsync_pol is only effective when tcmi_xsync_sat_sel = 0

tcmi_vsync_pol: VSYNC polarity
0: VSYNC active low
1: VSYNC active high

tcmi_hsync_pol: HSYNC polarity
0: HSYNC active low
1: HSYNC active high

tcmi_edge_sel: DCLK edge select to align all other TCMI outputs
0: falling edge
1: rising edge

#### 8.7.23. Temp_

| Temp_tl_cal1 | | Temperature sensor top-left calibration point 1 | 0xE8 |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_tl_cal2 | | Temperature sensor top-left calibration point 2 | 0xE9 |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_tr_cal1 | | Temperature sensor top-right calibration point 1 | 0xEA |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_tr_cal2 | | Temperature sensor top-right calibration point 2 | 0xEB |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_bl_cal1 | | Temperature sensor bottom-left calibration point 1 | 0xEC |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_bl_cal2 | | Temperature sensor bottom-left calibration point 2 | 0xED |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_br_cal1 | | Temperature sensor bottom-right calibration point 1 | 0xEE |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

| Temp_br_cal2 | | Temperature sensor bottom-right calibration point 2 | 0xEF |
|---|---|---|---|
| Operation | RW | | |
| Default | 0xFF | | |

#### 8.7.24. User_

| User_1 | | User register 1 for user data | 0xF0 |
|---|---|---|---|
| Operation | RW | | |
| Default | 0x00 | | |

| User_2 | | User register 2 for user data | 0xF1 |
|---|---|---|---|
| Operation | RW | | |
| Default | 0x00 | | |

#### 8.7.25. ENGINEERING_ID

| ENGINEERING_ID | Chip ID register SEG6 *: Engineering ID for MPWs | 0xF5 |
|---|---|---|
| Operation | R | |
| Default | according engineering ID | |

#### 8.7.26. WAFER_ID_

| WAFER_ID_MSB | Chip ID register SEG6 *: MSB of wafer ID | 0xF6 |
|---|---|---|
| Operation | R | |
| Default | according wafer ID | |

| WAFER_ID_LSB | Chip ID register SEG6 *: LSB of wafer ID | 0xF7 |
|---|---|---|
| Operation | R | |
| Default | according wafer ID | |

#### 8.7.27. CHIP_ID_

| CHIP_ID_MSB | Chip ID register SEG6 *: MSB of chip ID on wafer | 0xF8 |
|---|---|---|
| Operation | R | |
| Default | according chip ID | |

| CHIP_ID_LSB | Chip ID register SEG6 *: LSB of chip ID on wafer | 0xF9 |
|---|---|---|
| Operation | R | |
| Default | according chip ID | |

#### 8.7.28. PART_

| PART_TYPE | Chip ID register SEG6 *: Part type | 0xFA |
|---|---|---|
| Operation | R | |
| Default | according part ID | |

| PART_VERSION | Chip ID register SEG6 *: Part version | 0xFB |
|---|---|---|
| Operation | R | |
| Default | according part version | |

# 9. Addendum

### 9.1. Terms, Definitions and Abbreviations

| Abbreviation | Term, Definition | Explanation |
|---|---|---|
| ABS | Automatic Backlight Suppression | |
| ADC | Analog Digital Converter | |
| AMR | Ratio of ambient-light / modulated light | |
| Big-endian | | The MSB of a multi-byte register is stored in the memory location with the lowest address first. The next byte value in significance is stored at the following memory location and so on |
| CGU | Clock Generation Unit | |
| CSP | Chip Scale Package | |
| DCS | Differential Correlation Sample | |
| DLL | Delay Locked Loop | A digital circuit similar to PLL, with the main difference being the absence of an internal VCO, replaced by a delay line |
| HDR | High Dynamic Range | |
| IC | Integrated Circuit | |
| JTAG | Joint Test Action Group | |
| LED/LD | Light Emitting Diode / Laser Diode | |
| LSB | Least Significant Bit | |
| LSFR | Linear Feedback Shift Register | |
| MGA | Modulation Gate A | |
| MGB | Modulation Gate B | |
| MGX | Modulation Gate A, B | |
| mga | MGA control signal | |
| mgb | MGB control signal | |
| mgx | MGX control signal | |
| MSB | Most Significant Bit | |
| OSC | Oscillator | |
| PLL | Phase Locked Loop | |
| PN | Pseudo-random Noise | |
| ROI | Region of Interest | |
| QVGA | Quarter VGA | 320x240 pixel resolution |
| SGA | Storage Gate A | |
| SGB | Storage Gate B | |
| SGX | Storage Gate A, B | |
| TCMI | TOF Camera Module Interface | |
| TOF | Time of Flight | |
| VCO | Voltage Controlled OSC | |
| VGA | Video Graphics Array | 640x480 pixel resolution |
| XTAL | Crystal | |

Table 43: Definitions and Abbreviations

### 9.2. Related documents


- NXP I2C-bus specification: I2C Bus Specification and User Manual, NPX corp., Rev.5 - 9 October 2012
- Application note AN10 epc600/epc610 temperature and BG light compensation, ESPROS Photonics corp., 2014
- Handbook – epc 600 Time-of-flight range finder chip, ESPROS Photonics corp., 2014

## 10. IMPORTANT NOTICE

ESPROS Photonics AG and its subsidiaries (epc) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to epc's terms and conditions of sale supplied at the time of order acknowledgment.

epc warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with epc's standard warranty. Testing and other quality control techniques are used to the extent epc deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

epc assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using epc components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

epc does not warrant or represent that any license, either express or implied, is granted under any epc patent right, copyright, mask work right, or other epc intellectual property right relating to any combination, machine, or process in which epc products or services are used. Information published by epc regarding third-party products or services does not constitute a license from epc to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from epc under the patents or other intellectual property of epc.

Resale of epc products or services with statements different from or beyond the parameters stated by epc for that product or service voids all express and any implied warranties for the associated epc product or service. epc is not responsible or liable for any such statements.

epc products are not authorized for use in safety-critical applications (such as life support) where a failure of the epc product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of epc products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by epc. Further, Buyers must fully indemnify epc and its representatives against any damages arising out of the use of epc products in such safety-critical applications.

epc products are neither designed nor intended for use in military/aerospace applications or environments unless the epc products are specifically designated by epc as military-grade. Only products designated by epc as military-grade meet military specifications. Buyers acknowledge and agree that any such use of epc products which epc has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

epc products are neither designed nor intended for use in automotive applications or environments unless the specific epc products are designated by epc as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, epc will not be responsible for any failure to meet such requirements.



POLYTEC GmbH Polytec-Platz 1 - 7 D -76337 Waldbronn GERMANY
Tel: +49 (72 43) 604 1730 Fax: +49 (72 43) 6 99 44 E-Mail: oek@polytec.de www.polytec.de