General Description

TOF>range 611 is a TOF range-finder module based on the epc611 chip. It allows distance measurement in a very cost effective design from a few centimeters up to several meters.

The firmware is built to operate in a high dynamic range of up to 90dB. However, the lens is designed with aspheric lenses in order to reduce the required dynamic range as much as possible. A high dynamic range can be used even with a high measurement rate.

The TOF>range 611 comes fully calibrated, so the user can simply readout the clean distance data. The device is designed in a lightweight but very stable plastic frame which allows it to be mounted to a fast rotating turnable. Mounting posts on top and on bottom allow various mounting locations and concepts.

Features

- Up to 15 m distance measurement range on white target
- Measurement rates of more than 500 measurements per second possible
- Customer specific versions and functionality available upon request
- Very light weight
- Low power consumption
- Very robust and stable
- Ambient-light compensated up to 100 kLux
- Temperature compensated
- High speed serial interface
- Small detection spot 3 x 3 mm @ 1m

Applications

- Distance measurement from centimeters to meters
- Rotating scanners for scenery mapping (SLAM)
- Level measurement
- Object classification

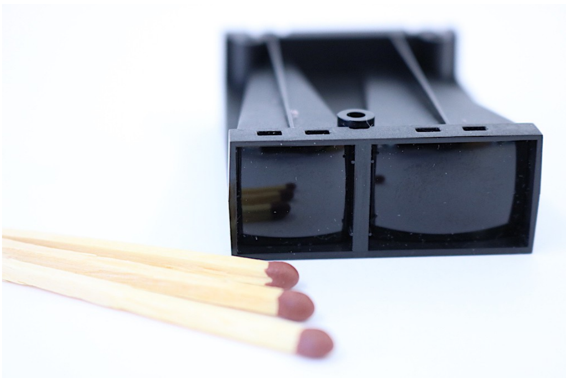

Figure 1: Product photo, front view

Block Diagram

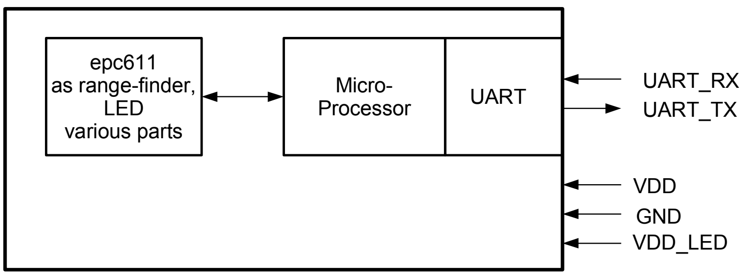

Figure 2: Block diagram

Table of contents

1. Technical data

T_A = 25°C, VDD = 5V, object reflectivity 90%, unless otherwise stated

Parameter	Description	Conditions	Min.	Typ	Max.	Units	Comments
VDD	Main supply voltage	Ripple [1] < 50 mV_{pp}	4.75	5.0	5.25	V	
VDDLED	LED supply voltage	Ripple [1] < 200 mV_{pp}	4.75	5	5.25	V	
IDD	Main supply current	Acquisition		17		mA	@ VDD 5V
		Power down		86		mA	@ VDD 5V
IDDLED	LED supply current	Acquisition		130		mA	Peak, @ VDDLED 5V
		Power down		9		mA	@ VDDLED 5V
D	Operating range	10 MHz mod. freq.	0.05		15.0	m	Measured with an object of at least the size of the spot d_{SPOT}, depends on integration time t_{INT} and object reflectivity.
		20 MHz mod. freq.	0.05		7.5		
FOVh/v	Horizontal field of view			0.18		°	
d_{SPOT}	Sensor spot size	d = 1.0m distance		3 x 3		mm	
		d = 2.0m distance		6 x 6			
		d = 3.5m distance		11 x 11			
		d = 5.0m distance		16 x 16			
		d = 7.0m distance		22 x 22			
		d = 10.0m distance		31 x 31			
Acc_{TRIM}	Accuracy calibrated and compensated, mean of 100 samples	0.2 ... 2.0 m		± 4		cm	Independent of the object reflectivity. Amplitude between 13k – 37 kLSB, 10 or 20MHz modulation frequency
		2.0 ... 7.5 / 15.0 m		± 2		%	
D_{NOISE}	Distance noise 1σ value	10 MHz		0.40		cm	Amplitude between 13k – 37 kLSB
		20 MHz		0.25			
t_{INT}	Integration time selectable		1		1'600	µs	Default 125µs
t_{CYCLE}	Measurement cycle time Refer to chapter 6.2, Timing.			1.9		ms	@ t_{INT} = 125µs
t_{WARM_UP}	Warm-up time until output data are in tolerance		Refer to chapter 6.6				
Res	Resolution			0.1		mm/LSB	
	Measurement output for 0.0 … 15.0m		0		150'000	LSB	
V_{APM}	Amplitude		18'000		131'000	LSB	Amplitude range for accurate results
T_A	Ambient temperature range (operation)		-20		85	°C	
T_{STO}	Storage temperature range		-20		85	°C	
Φ_L	Ambient-light			100		kLux	indirect, on target
RH	Relative humidity		15		90	%	Non condensing
ESD	Electrostatic discharge rating				2	kV	Human body model
	Weight			12.5		g	

Table 1: Technical data

Note:

[1] Min. and Max. voltage values include noise and ripple voltages.

2. Mechanical dimensions

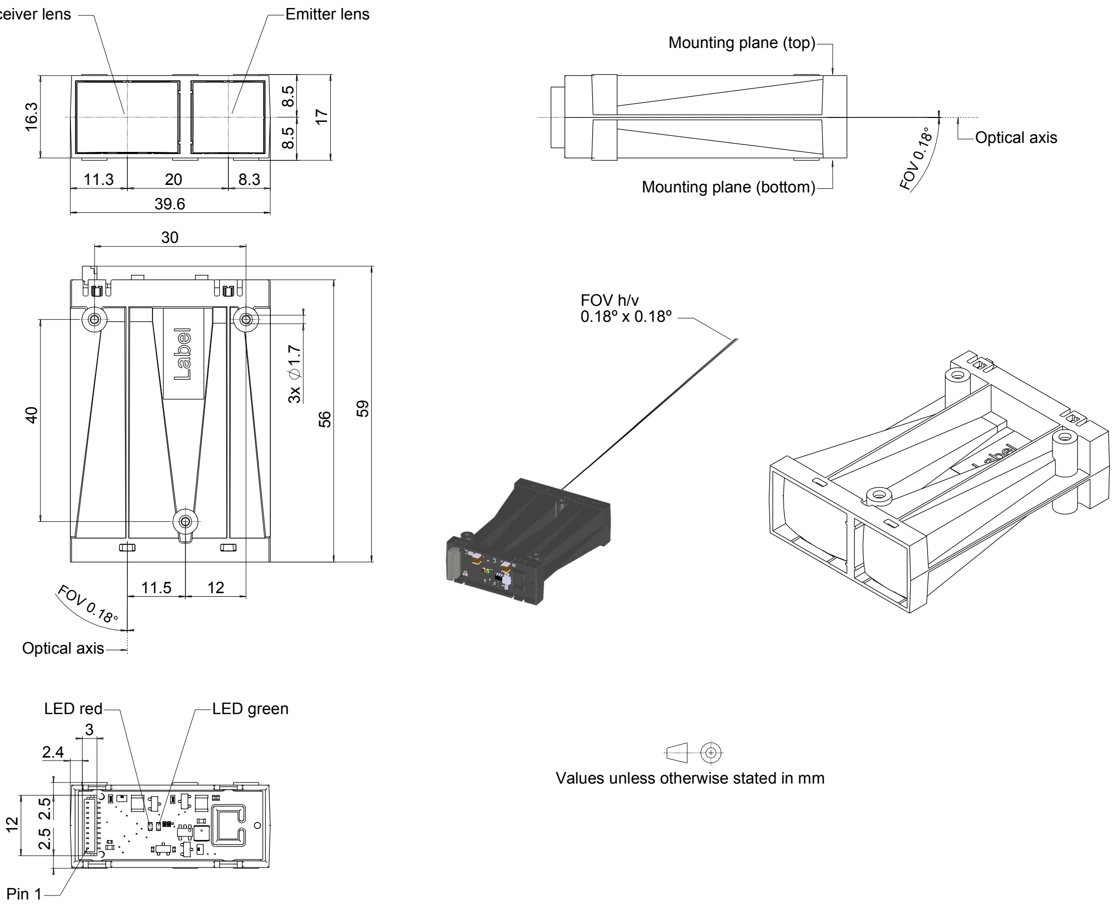

Figure 3: Mechanical dimensions

Notes:

- Zero value is for measured distances at the front of the housing.
- The TOF>range 611 is a very light-sensitive device but its main body is not totally hermetic for stray-light from sides or backside. Therefore for achieving best performance in ambient-light environments, it is suggested to build it into a stray-light dense application housing which is usually anyway the case.

3. Sensor interface

3.1. Connection

Connector type: JST BM10B-SRSS-TB(LF)(SN), 10 pin

Figure 4: Connector

3.2. Pin table

No.	Name	Function	Comments
1	VDD	Main supply voltage	Stable and free of noise. Decouple with min. 10µF low ESR capacitor to GND
2	GND	Ground	
3	PIN3	reserved	Do not connect
4	PIN4		
5	PIN5		
6	UART_TX	Data output Tx	UART interface, LVTTL levels (3.3V)
7	UART_RX	Data input Rx	
8	PIN8	no function	Do not connect
9	GND	Ground	
10	VDDLED	LED supply voltage	Stable and free of noise. Decouple with min. 10µF low ESR capacitor to GND

Table 2: Pin list

3.3. Communication protocol

The communication interface uses 8 bit UART standard on LVTTL levels (3.3V) It operates with following parameters:

Parameter	Value	Unit	Remark
Baud rate	921'600	Bit/s	1 bit = 1.085µs
Start bits	1	Bit	low active
Data	8	Bit	
Stop bits	1	Bit	high active
Parity	No		

Table 3: UART configuration

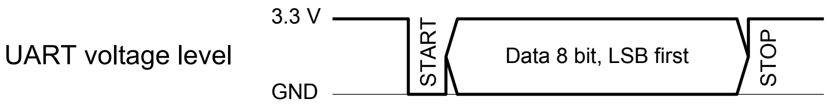

Figure 5: UART frame format

The UART operates in a master-slave mode with the application as the master and the TOF>range 611 as the slave.

A request is initiated with a command of the master by polling. The TOF>range 611 as the slave returns the answer to the request after the internal processing time t_{PROC}.

TOF>range 611 does not accept commands during the processing t_{PROC} and the communication t_{COM_TX}. A next command can be issued earliest after finishing Data Out.

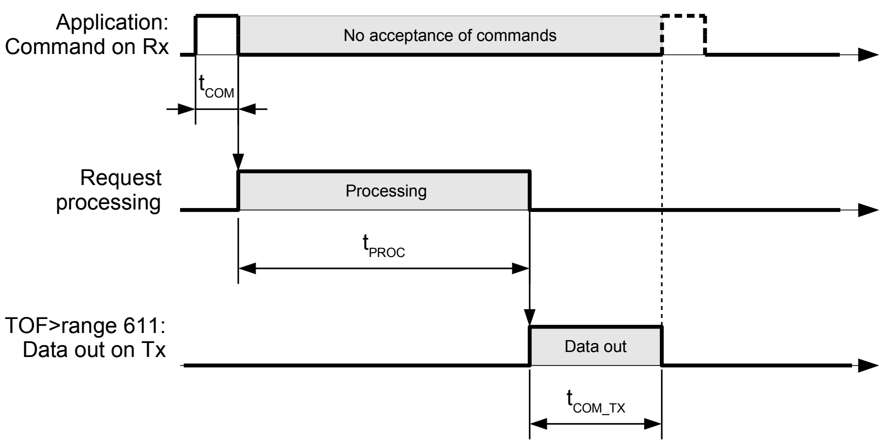

Figure 6: Command and answer sequence

4. System Boot-up

Apply power: VDD and VDDLED. The sequence does not matter. The device notifies the power-up with a short red LED flash and is now in the power down mode. After sending the command for power-up, the green LED turns on and the device is ready.

Error cases:

- Red LED flashing: Firmware not correctly downloaded. Download the firmware anew with the GUI of the epc611 evaluation kit or with the bootloader (refer to chapter 5.20 and 5.27).
- Red LED stays on: Error during boot-up. Switch off/on power again.
- If the error remains, contact your sales responsible.

5. Commands and responses

TOF>range 611 answers to each command with either the required data, acknowledge, not acknowledge or an error. LSByte is transmitted first, MSByte last. Use only commands listed.

5.1. Command format

The command packet has a fixed length of 14 Bytes: A start byte (value 0xF5), followed by 1 byte command ID (CMD), 8 bytes of parameters corresponding to the command and 4 closing bytes with a 32bit CRC.

1 byte	1 byte	8 bytes	4 byte
0xF5	**CMD**	**Parameters, fix length**	32 bit CRC
		LSByte … MSByte	LSByte … MSByte

Figure 7: Command format

5.2. Response format

The answer packet has variable length: A start byte (value 0xFA), followed by 1 byte type definition, 2 byte length definition n, n bytes data and 4 closing bytes with a 32bit CRC.

1 byte	1 byte	2 bytes	n bytes	4 bytes
0xFA	**Type**	**Length n**	**Data**	32 bit CRC
		LSByte MSByte	LSByte … MSByte	LSByte … MSByte

Figure 8: Response format

5.3. CRC calculation

The Cyclic Redundancy Check (CRC) calculation includes all bytes of the packet except the CRC itself. Examples are listed in the command list.

CRC specification:

- Bytewise CRC32
- Init value: 0xFFFFFFFF
- Xor value: 0x00000000
- Polynom: 0x04C11DB7

CRC calculation function:

```
uint32_t CrcCalc::calcCrc32(const uint8_t *data, const uint32_t size)
{
  uint32_t crc = initValue;
  for(uint32_t i = 0; i < size; i++)
  {
    crc = calcCrc32Uint8(crc, data[i]);
  }
  return crc ^ xorValue;
}

uint32_t CrcCalc::calcCrc32Uint8(uint32_t crc, uint8_t data)
{
  int32_t i;
  //This shift is done to make it compatible to the STM32 hardware CRC
  crc = crc ^ (data << 24);
  for (i = 0; i < 8; i++)
  {
    if (crc & 0x80000000)
    {
      crc = (crc << 1) ^ polynom;
    }
    else
```

```
    {
      crc = (crc << 1);
    }
  }
  return (crc);
}
```

5.4. SET_POWER [0x40]

Puts the device in power-up or power-down mode and returns after t_{PROC} acknowledge. Before any acquisition commands, power must be enabled. The green LED indicates power on.

Parameter byte 0: 0x00 disables power, 0x01 enables power
others: 0x00

Command e.g. | 0xF5 | **0x40** | **0x01** 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x9C 0xD7 0xD6 0x91 | (power enable)

Response type 0x00: Acknowledge

Response data 0 bytes

Response time power up t_{PROC} : < 200 ms

Response time power down t_{PROC} : < 30 µs

Response e.g | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.5. SET_MODULATION_FREQUENCY [0x05]

Sets the modulation frequency and returns after t_{PROC} acknowledge. The modulation frequency is stored and used as long power is on or until a new value is set. Default at startup is 10MHz.

Parameter byte 0: 0x00 = 10 MHz; 0x01 = 20 MHz
others: 0x00

Command e.g. | 0xF5 | **0x05** | **0x01** 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0xCF 0x9D 0x83 0xC7 | (20 MHz)

Response type 0x00: Acknowledge

Response data 0 bytes

Response time t_{PROC} : ~ 45 µs

Response e.g. | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.6. SET_INTEGRATION_TIME_DIS [0x00]

Sets the integration time for distance measurements and returns after t_{PROC} acknowledge. The integration time is stored and used as long power is on or until a new time is set. Range: 1 … 1'600 µs. After power-up, default is 125 µs.

Parameter byte 1, 2: Integration time in microseconds, 16 bit unsigned integer
others: 0x00

Automatic Mode Integration time = 0: The device selects the integration time automatically. Refer to chapter 6.1.

Command e.g. | 0xF5 | **0x00** | 0x00 **0x1E 0x00** 0x00 0x00 0x00 0x00 0x00 | 0xD9 0x85 0x1A 0x99 | (30µs integration time)

Response type 0x00: Acknowledge

Response data 0 bytes

Response time t_{PROC} : ~ 40 µs

Response e.g | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.7. GET_INTEGRATION_TIME_DIS [0x27]

Returns after t_{PROC} the selected or used integration time for distance measurements.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x27** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0xC4 0x3F 0x68 0x4C |

Response type 0x09: Data

Response data 2 bytes: Integration time in microseconds, 16 bit unsigned integer

Response time ~ 40 µs

Response e.g. | 0xFA | **0x09** | **0x02 0x00** | **0x5E 0x01** | 0x83 0xF9 0x91 0xF0 | (350 µs)

5.8. GET_DISTANCE [0x20]

Starts a new distance acquisition and returns after t_{PROC} the result or status.

Parameter no, all bytes 0x00

Command e.g | 0xF5 | **0x20** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x98 0x53 0xE9 0x9B |

Response type 0x03: Data

Response data 4 bytes: Distance, 0.1mm/LSB or status, 32 bit unsigned integer. Zero according the note of Figure 3.
Readout order: Starts at row 0, pixel 0 … pixel 7 and ends with row 7, pixel 7.
Ranges: 10 MHz modulation frequency, 0 … 15m: 0 … 150'000d
20 MHz modulation frequency, 0 … 7.5m: 0 … 75'000d
Status: 16'001'000d: Low TOF amplitude
16'002'000d: ADC overflow
16'003'000d: Saturation
16'004'000d: Reserved
16'005'000d: ADC underflow
16'006'000d: High TOF amplitude

Response time Refer to chapter 6.2, Timing.

Response e.g. | 0xFA | **0x03** | **0x04 0x00**| **0xE8 0x04 0x00 0x00** | 0x14 0x97 0x4E 0xE1 | (e.g distance 125.6 mm)

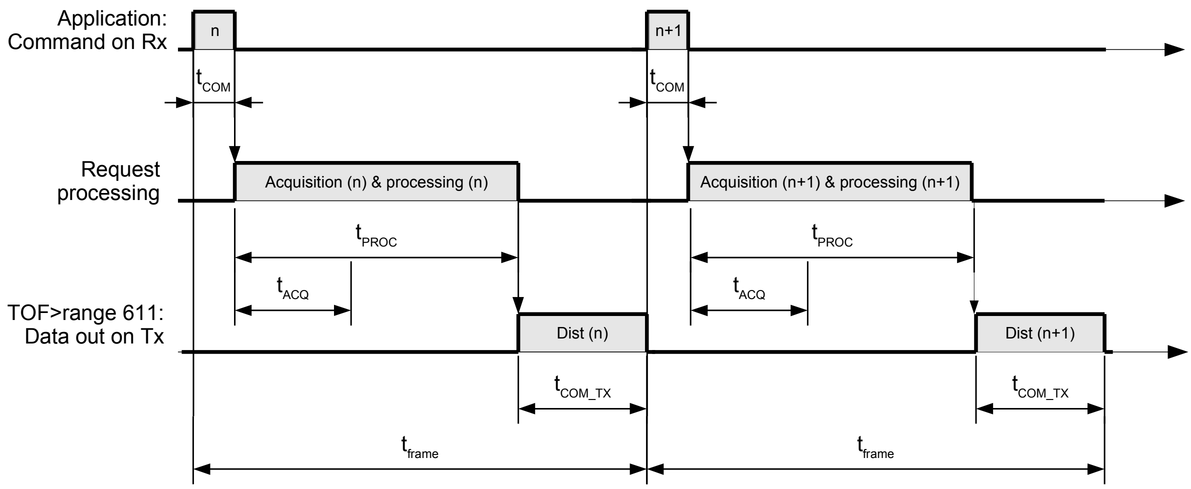

Figure 9: Timing of the distance measurements

5.9. GET_DISTANCE_AMPLITUDE [0x22]

Starts a new distance and TOF amplitude acquisition and returns after t_{PROC} the result or status. The amplitude is a quality indicator for the distance result.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x22** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0xE3 0x1A 0x29 0x7B |

Response type 0x05: Data

Response data 4 bytes distance followed 4 bytes amplitude

Distance byte 0..3: Distance, 0.1mm/LSB or status, 32 bit unsigned integer. Zero according the note of Figure 3.
Ranges: 10 MHz modulation frequency, 0 … 15m: 0 … 150'000d
20 MHz modulation frequency, 0 … 7.5m: 0 … 75'000d

Amplitude byte 0..3 : TOF amplitude or status, 32 bit unsigned integer.
Status: Values < 13'000 LSB, distance noise is significant; > 100'000 LSB, the distance can contain considerable error.
16'001'000d: Low TOF amplitude
16'002'000d: ADC overflow
16'003'000d: Saturation
16'004'000d: Reserved
16'005'000d: ADC underflow
16'006'000d: High TOF amplitude

Response time Refer to chapter 6.2, Timing. Refer to Figure 9 for the timing diagram.

Response e.g. | 0xFA | **0x05** | **0x08 0x00** | **0xD3 0x04 0x00 0x00 0x89 0x81 0x00 0x00** | 0x88 0x36 0x4A 0x63 |
(distance 123.5mm and amplitude 33'161 LSB)

5.10. GET_DCS [0x25]

Starts a new acquisition and returns after t_{PROC} DCS values only.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x25** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0xBF 0x76 0xA8 0xAC |

Response type 0x07: Data

Response data 4 bytes DCS0 followed by 4 bytes DCS1, 4 bytes DCS2 and 4 bytes DCS3

byte 0..3: DCSx, 32 bit 2's complement signed integer. Range: -131'072 … +131'071 LSB

0x 00 00 1F FF: Saturation

0x 00 00 1F FE: ADC overflow

0x FF FE 00 00: ADC underflow

Response time Refer to chapter 6.2, Timing. Refer to Figure 9 for the timing diagram.

Response e.g. | 0xFA | **0x07** | **0x10 0x00** | **0xDC 0x65 0x00 0x00 0x57 0x54 0x00 0x00 0xD4 0x9E 0xFF 0xFF 0x57 0xAE 0xFF 0xFF** | 0x09 0xFC 0xA4 0xCF | (DCS0 = 26'076, DCS1 = 21'591, DCS2 = -24'876, DCS3 = -20'905)

5.11. GET_DCS_DISTANCE_AMPLITUDE [0x23]

Starts a new acquisition and returns after t_{PROC} DCS, distance and amplitude values as a set.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x23** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x85 0xB0 0x29 0x89 |

Response type 0x08: Data

Response data 4 bytes DCS0 followed by 4 bytes DCS1, 4 bytes DCS2, 4 bytes DCS3, 4 bytes distance and 4 bytes amplitude

DCSx byte 0..3: DCSx, 32 bit 2's complement signed integer. Range: -131'072 … +131'071 LSB

0x 00 00 1F FF: Saturation

0x 00 00 1F FE: ADC overflow

0x FF FE 00 00: ADC underflow

Distance byte 0..3: Distance, 0.1mm/LSB or status, 32 bit unsigned integer. Zero according the note of Figure 3.

Ranges: 10 MHz modulation frequency, 0 … 15m: 0 … 150'000d

20 MHz modulation frequency, 0 … 7.5m: 0 … 75'000d

Amplitude byte 0..3: TOF amplitude or status, 32 bit unsigned integer.

Status: Values < 13'000 LSB, distance noise is significant; > 100'000 LSB, the distance can contain considerable error.

16'001'000d: Low TOF amplitude

16'002'000d: ADC overflow

16'003'000d: Saturation

16'004'000d: Reserved

16'005'000d: ADC underflow

16'006'000d: High TOF amplitude

Response time Refer to chapter 6.2, Timing. Refer to Figure 9 for the timing diagram.

Response e.g. | 0xFA | **0x08** | **0x18 0x00** | **0x6F 0x65 0x00 0x00 0x83 0x54 0x00 0x00 0x2D 0x9F 0xFF 0xFF 0x28 0xAE 0xFF 0xFF 0xB8 0x04 0x00 0x00 0x67 0x81 0x00 0x00** | 0x94 0xFC 0x43 0xBD |
(DCS0 = 25'967, DCS1 = 21'635, DCS2 = -24'787, DCS3 = -20'952, distance = 120.8mm, amplitude = 33'127)

5.12. GET_TEMPERATURE [0x4A]

Returns after t_{PROC} the chip temperature during last distance acquisition. Temperature information is useful for error compensation of the measured distance.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x4A** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x18 0x41 0xF5 0xA4 |

Response type 0xFC: Data

Response data 2 bytes: Temperature, 0.01 °C / LSB, 16 bit 2's complement signed integer.

Response time t_{PROC}: ~ 40 µs

Response e.g. | 0xFA | **0xFC** | **0x02 0x00** | **0x47 0x13** | 0x4F 0xEE 0x12 0x1F | (temperature 49.35°C)

5.13. DRNU_COMPENSATION [0x41]

Enables or disables the Distance Response Non-Uniformity (DRNU) compensation and returns after t_{PROC} acknowledge.

Parameter byte 0: 0x00 = Compensation enabled; 0x01 = Compensation disabled

others: 0x00

Command e.g. | 0xF5 | **0x41** | **0x01** 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0xFA 0x7D 0xD6 0x63 | (disable compensation)

Response type	0x00: Acknowledge
Response data	0 bytes
Response time	t_{PROC} : ~ 40 µs
Response e.g.	| 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.14. GET_FIRMWARE_VERSION [0x49]

Returns after t_{PROC} the firmware version of the TOF>range 611: Version.Subversion (V.s)

Parameter	no, all bytes 0x00
Command e.g.	| 0xF5 | **0x49** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x05 0xA2 0x35 0xB6 |
Response type	0xFE: Data
Response data	4 bytes byte 0, 1: Subversion s, 16 bit unsigned integer byte 2, 3: Version V, 16 bit unsigned integer
Response time	t_{PROC}: ~ 40 µs
Response e.g.	| 0xFA | **0xFE** | **0x04 0x00** | **0x0E 0x00 0x01 0x00** | 0xDA 0xD7 0x3A 0xFB | (Version 1.14)

5.15. GET_CHIP_INFORMATION [0x48]

Returns after t_{PROC} the epc611 Chip ID and Wafer ID.

Parameter	no, all bytes 0x00
Command e.g.	| 0xF5 | **0x48** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x63 0x08 0x35 0x44 |
Response type	0xFD: Data
Response data	4 bytes byte 0, 1: Chip ID, 16 bit unsigned integer byte 2, 3: Wafer ID, 16 bit unsigned integer
Response time	t_{PROC}: ~ 40 µs
Response e.g.	| 0xFA | **0xFD** | **0x04 0x00** | **0x10 0x04 0x10 0x00** | 0x4F 0x56 0xF8 0x21 | (Chip ID 1040 , Wafer ID 16)

5.16. GET_PROD_DATE [0x50]

Returns after t_{PROC} the production date of the TOF>range 611.

Parameter	no, all bytes 0x00
Command e.g.	| 0xF5 | **0x50** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x8B 0x10 0x32 0xD2 |
Response type	0xF9
Response data	2 bytes byte 0: Last two digits of the year as unsigned integer e.g. 18 byte 1: Number of the week as integer e.g. 22
Response time	~ 40 µs
Response e.g.	| 0xFA | **0xF9** | **0x02 0x00** | **0x12 0x16** | 0x00 0x76 0x04 0xA7 | (year 18, week 22)

5.17. DATA_NACK

System response only: Command not accepted or unknown.

Response type	0x01: Not acknowledged
Response data	0 bytes
Response e.g.	| 0xFA | **0x01** | **0x00 0x00** | (**0 bytes**) | 0x35 0x07 0x24 0xE9 |

5.18. DATA_ERROR

System response only: Error occurred during the execution of the command. Response instead of the required data

Response type	0xFF: Error
Response data	2 bytes: bit 0..14: Error number. Try it again. If the error remains, contact your sales responsible. bit 15: 0
Response e.g.	| 0xFA | **0xFF** | **0x02 0x00** | **0x03 0x00** | 0x94 0xF6 0x35 0x81 | (error No. 3)

5.19. IDENTIFY [0x47]

Returns after t_{PROC} the device identification ID and the mode (normal operation or bootloader e.g. a firmware update was not successful). Is TOF>range 611 in bootloader mode, run a firmware update with the epc611 evaluation kit GUI or with the bootloader (see next). The GUI detects missing firmware and runs update automatically. This command may be used for communication check.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x47** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x0A 0x67 0xF6 0x1D |

Response type 0x02: Data

Response data 4 bytes:

- byte 0: Hardware version
- byte 1: Device type, TOF>range 611 = 0x00
- byte 2: Chip type, epc611 = 0x06
- byte 3: 0x00 = normal operation, 0x80 = bootloader

Response time ~ 40 µs

Response e.g. | 0xFA | **0x02** | **0x04 0x00** | **0x00 0x00 0x06 0x00** | 0x0C 0x81 0x5B 0x28 | (HW version 0, TOF>range 611, epc611, normal operation)

| 0xFA | **0x02** | **0x04 0x00** | **0x00 0x00 0x06 0x80** | 0xE2 0x61 0x57 0x41 | (HW version 0, TOF>range 611, epc611, bootloader)

5.20. JUMP_TO_BOOTLOADER [0x44]

Stops all normal operation activities and changes to bootloader. Now, the bootloader is answering to this and all following commands. Refer also to chapter 5.27.

Parameter no, all bytes 0x00

Command e.g. | 0xF5 | **0x44** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x17 0x84 0x36 0x0F |

Response type 0x00: Acknowledge

Response data 0 bytes

Response time t_{PROC} : < 5ms

Response e.g. | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.21. UPDATE_FIRMWARE [0x45]

Bootloader command only: Copies the firmware into the flash memory of the sensor. It returns acknowledge after t_{PROC}.

Procedure:
1st, write control byte “start” with password and file size; 2nd, write control byte “write” with index and data; 3rd, write control byte “complete”.

Parameter 8 bytes: Contents differs and depends on operation step: Refer to Table 4.

Step	Action	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
1st	Start	Size of the update file				Password = 0x654321			**0x00**
2nd	Write data	Firmware data (4 bytes)				FirmwareData[index] (3 bytes)			**0x01**
3rd	Complete	All 7 bytes = 0x00							**0x02**

Table 4: Bootloader data format

Command e.g.
| 0xF5 | **0x45** | **0x00 0x21 0x43 0x65 0x10 0x00 0x00 0x00** | 0x29 0x7B 0xFA 0x1C | (start for 16 byte file size)
| 0xF5 | **0x45** | **0x01 0x00 0x00 0x00 0x10 0x4A 0x56 0x50** | 0x10 0x20 0x8A 0xE6 | (write data to index 0)
| 0xF5 | **0x45** | **0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00** | 0xE5 0x97 0x75 0x4A | (complete)

Response type 0x00: Acknowledge

Response data 0 bytes

Response time t_{PROC} : < 400ms

Response e.g. | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.22. WRITE_CALIBRATION_DATA [0x4B]

Writes the calibration data into the flash memory and returns after t_{PROC} acknowledge. This data are used during DRNU compensation. Warning: Deletes previous stored calibration.

Procedure:

1st, write control byte "start" with password and file size; 2nd, write control byte "write" with index and data; 3rd, write control byte "complete".

Parameter 8 bytes: Contents differs and depends on operation step: Refer to Table 5.

Step	Action	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
1st	Start	Size of the update file				Password = 0x654321			**0x00**
2nd	Write data	Calibration data (4 bytes)				CalibrationData[index] (3 bytes)			**0x01**
3rd	Complete	All 7 bytes = 0x00							**0x02**

Table 5: Calibration data download format

Command e.g. | 0xF5 | **0x45** | **0x00 0x21 0x43 0x65 0x10 0x00 0x00 0x00** | 0x29 0x7B 0xFA 0x1C | (start for 16 byte file size)
| 0xF5 | **0x45** | **0x01 0x00 0x00 0x00 0x10 0x4A 0x56 0x50** | 0x10 0x20 0x8A 0xE6 | (write data to index 0)
| 0xF5 | **0x45** | **0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00** | 0xE5 0x97 0x75 0x4A | (complete)

Response type 0x00: Acknowledge
Response data 0 bytes
Response time t_{PROC} : < 400ms
Response e.g. | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.23. SET_DLL_STEP [0x06]

Sets the number of DLL steps for artificial phase/distance shifting and returns after t_{PROC} acknowledge. 1 step is around 2.15ns / 315mm.

Parameter byte 0: Number of DLL steps
others: 0x00
Command e.g. | 0xF5 | **0x06** | **0x01** 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0xD2 0x7E 0x43 0xD5 | (Number of steps = 1)

Response type 0x00: Acknowledge
Response data 0 bytes
Response time t_{PROC} : ~ 40 µs
Response e.g. | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |

5.24. WRITE_REGISTER [0x4C]

Writes the value into a register of the sensor and returns after t_{PROC} 16bit SPI response from epc611 chip.

Parameter 3 bytes:
byte 0: Register address inside the page (0x00 … 0x20)
byte 1: Page address
byte 2: Register value
others: 0x00
Command e.g. | 0xF5 | **0x4C** | **0x01 0x00 0x56** 0x00 0x00 0x00 0x00 0x00 | 0x7D 0xAD 0xE1 0xE6 |
(Write into page 0, register 1 = 0x56)

Response type 0xFB: Data
Response data 2 bytes: 16bit SPI response from epc611 chip
Response time t_{PROC} : ~ 40 µs
Response e.g. | 0xFA | **0xFB** | **0x02 0x00** | (**2 bytes**) | CRC (4 bytes) |

5.25. READ_REGISTER [0x4D]

Reads the value of the sensor register and returns after t_{PROC} 16bit SPI response from epc611 chip.

Parameter 2 bytes:
byte 0: Register address inside the page (0x00 … 0x20)
byte 1: Page address
others: 0x00
Command e.g. | 0xF5 | **0x4D** | **0x01 0x00** 0x00 0x00 0x00 0x00 0x00 0x00 | 0x8E 0xF1 0xD5 0x28 | (read page, 0, register 1)

Response type 0xFB: Data
Response data 2 bytes: 16bit SPI response from epc611 chip
Response time t_{PROC} : ~ 40 µs
Response e.g. | 0xFA | **0xFB** | **0x02 0x00** | **(2 bytes)** | CRC (4 bytes) |

5.26. READ_NOP [0x4E]

Returns after t_{PROC} the value of the last SPI command to the epc611.

Parameter	no, all bytes 0x00
Command e.g.	| 0xF5 | **0x4E** | 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x59 0xCE 0xB4 0x61 |
Response type	0xFB : last 16 bit SPI command to the epc611
Response data	2 byte
Response time	t_{PROC} : ~ 40 µs
Response e.g.	| 0xFA | **0xFB** | **0x02 0x00** | (2 **bytes)** | CRC (4 bytes) |

5.27. Firmware update

A firmware update of the TOF>range 611 in the field is possible. In such a case, ESPROS provides a valid update file on their website in the ESPROS epc611 Evaluation Kit Software package. The upgrade will neither touch nor overwrite the calibration with the bootloader. The bootloader is bounded on 12 kByte Flash Memory for the update file. Calibration data has additionally own 2 kByte.

A certain sequence of commands needs to be executed for uploading the file to the device:

Update file, used in the table below (no valid file, example only)
0x10 0x4A 0x56 0x50 0xFF 0x67 0xA0 0xC0
0x23 0x45 0xAA 0x00 0x34 0x78 0x99 0xBB

Host action	Device reaction	Device answer
Send command "JUMP_TO_BOOTLOADER": | 0xF5 | **0x44** | **0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00** | 0x17 0x84 0x36 0x0F |	Jump to bootloader	ACK *
Send command "UPDATE_FIRMWARE" with control byte "0", the password and the size of the update file: | 0xF5 | **0x45** | **0x00 0x65 0x43 0x21 0x10 0x00 0x00 0x00** | 0xEE 0xBF 0xD0 0x97 |	Verify password and store file size	ACK *
Send the command "UPDATE_FIRMWARE" with control byte "1" , the index and 4 bytes of the update file. Repeat this step as often as needed, update file size / 4, e.g. with given update file above: | 0xF5 | **0x45** | **0x01 0x00 0x00 0x00 0x10 0x4A 0x56 0x50** | 0x10 0x20 0x8A 0xE6 | | 0xF5 | **0x45** | **0x01 0x04 0x00 0x00 0xFF 0x67 0xA0 0xC0** | 0xA1 0x67 0xFB 0xA7| | 0xF5 | **0x45** | **0x01 0x08 0x00 0x00 0x23 0x45 0xAA 0x00** | 0x7A 0x71 0x34 0xA7| | 0xF5 | **0x45** | **0x01 0x0C 0x00 0x00 0x34 0x78 0x99 0xBB** | 0x8B 0x89 0xB2 0x8F|	Store data Store data Store data Store data	ACK * ACK * ACK * ACK *
Send command "UPDATE_FIRMWARE" with control byte "2": | 0xF5 | **0x45** | **0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00** | 0xE5 0x97 0x75 0x4A |	Verify data and return to normal operation	ACK *
Wait to exceed boot time. The device is now ready to operate. Communication may be tested with the command "IDENTIFY"	-	-

Table 6: Update procedure

Notes:
- * ACK: ACKNOWLEDGE: Response e.g.: | 0xFA | **0x00** | **0x00 0x00** | (**0 bytes**) | 0xB2 0xAB 0xFC 0xE8 |
- If an error occurs (e.g. corrupted data, invalid command), the device answers with "NOT ACKNOWLEDGE".
- Is the update procedure interrupted, no valid application is in the flash. The device stays in bootloader mode.
- In such cases, the update procedure must be restarted. It can be repeated at any time.

6. Application information

6.1. Automatic Mode

It will be enabled by setting the integration time = 0. If chosen, the TOF>range 611 selects automatically an adequate integration time for getting most reliable results. It uses the following simple algorithm which will be done after every distance acquisition:

```
result = 35'000 – TOF_Amplitude

if (result > 2'000){
  newIntegrationTime = (integrationTime * 12) / 10
}
else if(result < -2'000){
  newIntegrationTime = (integrationTime * 8) / 10
}
```

GET_INTEGRATION_TIME_DIS [0x27] reads back the integration time selected by TOF>range 611. In automatic mode, it will read back the integration time of the next distance acquisition

6.2. Timing

The timings for distance, amplitude and DCS acquisition are

t_{int} < 650 µs :→t_{frame} = 4 * t_{int} + 860µs (acquisition + grayscale time) + 150µs (calculation time) + 210µs (t_{COM})+ 180µs / 240µs (t_{COM_TX})

t_{int} > 650 µs :→t_{frame} = t_{int} * 10 (to not exceed their maximum power limit of illumination)

Command	t_{PROC}	t_{frame}	Unit
GET_DISTANCE	1.51	1.90	ms
GET_DISTANCE_AMPLITUDE	1.51	1.96	ms

Table 7: Examples for t_{frame} @ t_{INT} = 125µs

6.3. Operating range (unambiguity distance)

The TOF>range 611 uses the continuous wave TOF phase-shift measurement technique with a modulation frequency of 10 / 20 MHz. This leads to an unambiguity distance of 15 / 7.5 m. Highly reflective objects at a distance greater than 15 / 7.5 m will appear closer due to wrap-around of the modulation period.

Example @ 20MHz: With a highly reflective object at a distance of 8.5m, the output of the TOF>range 611displays 1m (8.5m – 7.5m). This is obviously wrong.

6.4. Object reflectivity

The reflectivity of the object can have an impact on the distance accuracy. Such a distance error can be corrected by an additional compensation based on measuring the amplitude.

6.5. Measurement example

The following figure shows a typical example result of a distance measurement from a TOF>range 611 pointing to a white object.

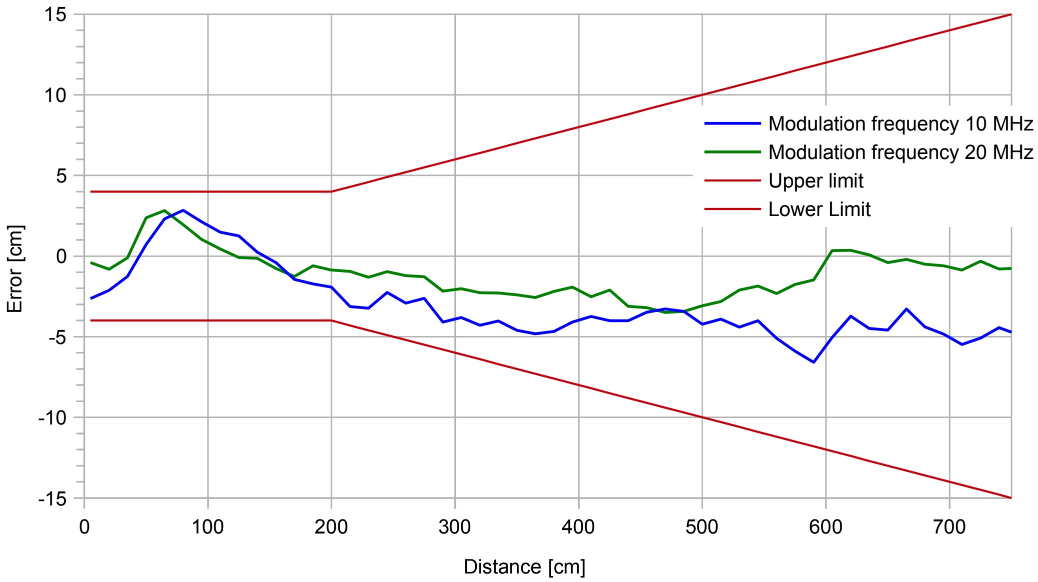

Figure 10: Typical measurement error versus distance

6.6. Warm-up distance drift

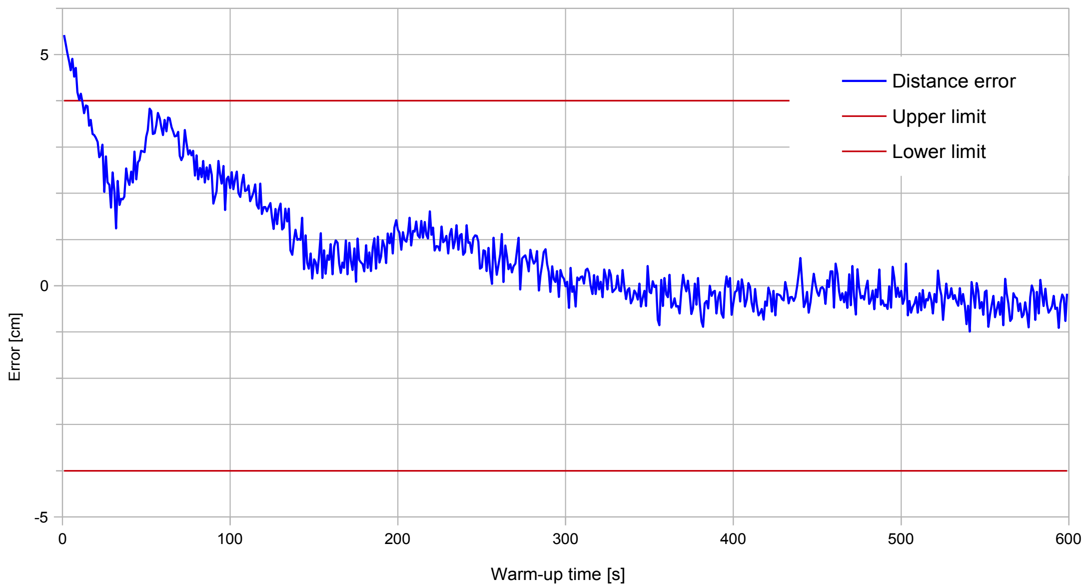

Figure 11: Typical warm-up phase of TOF>range 611

7. Ordering Information

Part Number	Part Name	Comment
P100 483	TOF>range 611	
P100 498	USB-UART Adapter	Refer to epc611 Evaluation Kit
P100 516	Cable 10 Pin F JST 1.0 mm (L=76 mm)	Refer to epc611 Evaluation Kit.
P100 487	epc611 Evaluation Kit	

Table 8: Ordering Information

IMPORTANT NOTICE

ESPROS Photonics AG and its subsidiaries (ESPROS) 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 ESPROS' terms and conditions of sale supplied at the time of order acknowledgment.

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

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

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

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

ESPROS products are not authorized for use in safety-critical applications (such as life support) where a failure of the ESPROS 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 ESPROS products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by ESPROS. Further, Buyers must fully indemnify ESPROS and its representatives against any damages arising out of the use of ESPROS products in such safety-critical applications.

ESPROS products are neither designed nor intended for use in military/aerospace applications or environments unless the ESPROS products are specifically designated by ESPROS as military-grade. Only products designated by ESPROS as military-grade meet military specifications. Buyers acknowledge and agree that any such use of ESPROS products which ESPROS 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.

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