ToF10120 Time-of-flight ranging sensor, user manual.

Author: W. Nelis
Date: 2024.07.16

This is a translation of a part of the data sheet of the ToF10210 sensor, which is written partly in English and partly in Chinese. The part specifying the API however is for the major part written in Chinese. Below is an English translation, including a little editing on my part. The chapter numbering of the data sheet is retained, but some extra chapters are added with information from other sources and personal observations.

Two different versions of the data sheet of the ToF10120 sensor were found, with different information but without a version number. Below the information that can be found in at least one of those versions is bundled and thus some of the information might be outdated.

2.3. Serial port (UART) communication parameters

Communication speed:	9600 [b/s]
Data bit count:	8
Parity:	None
Stop bit count:	1
Flow control method:	None

2.4. API

In this chapter the API for both serial communication via a UART and communication via an I2C bus are specified.

2.4.1. UART reading

The table below specifies the parameters which can be retrieved, the string to be send to the sensor to obtain the parameter and the format of the return value.

Description	Send	Return	Remarks
Offset	r1#	D=xxmm	xx= 0~99 mm, default 0 mm
Serial port send interval	r2#	T=xxxxmS	xxxx=10~9999 ms, default 100 ms
Distance mode	r3#	M=x	0=filtered distance, 1=real-time distance, default 0
Maximum distance	r4#	Max=xxxxmm Max>2000mm	xxxx=10~2000 mm, default no limit
Medium mode	r5#	S=x	0=active send via UART, 1=passive read via UART or I2C, default 0
Distance	r6#	L=xxxxmm	xxxx= 0~2000 mm, value is only valid if in passive read mode
I2C address	r7#	I=xxx	xxx= 1~254, default 164 == 0xA4

XTAL calibration parameter *(‡)*	r8#	X=xxx	xxx= 0~200, 0 before calibration

(‡) : This parameter does not seem to exist in all sensors.

2.4.2. UART writing

The table below specifies the format of the command to send to the sensor in order to modify a parameter and the allowable range for the value of the parameter. In all cases a line is returned with the status of the write operation: ‘ok!’ in case it was successful, ‘fail’ if the write was not successful. The latter might be caused by a parameter value which is out of range.

Description	Send	Remarks
Update offset	s1+xx# s1-xx#	Increase (+xx) or decrease (-xx) the offset. A value of either ‘+0’ or ‘-0’ sets the offset to zero.
Serial port send interval	s2-xxxx#	Xxxx=10~9999 ms, default 100 ms
Distance mode	s3-x#	0=filtered distance, 1=real-time distance, default 0
Maximum distance	s4-xxxx#	xxxx=10~2000 mm. A value of 0 sets the maximum to unlimited.
Medium mode	s5-x#	0=active send via UART, 1=passive read via UART or I2C
I2C address	s7-xxx#	xxx= 1~254, default 164 == 0xA4
Calibration *(‡)*	s8-x#	Calibration successful: 0=returns offset deviation value, 1=returns xtalk deviation parameter.

(‡) : This parameter does not seem to exist in all sensors.

2.4.2.1. Various remarks

Using the TOF10120 sensor via the UART interface, the following characteristics were observed:

- It seems that any command on the serial interface must be entered within a short time frame. All attempts to enter a command by hand via a terminal program, such as minicom, failed.
- Once a parameter is set using the serial interface with an s-command it remains set. The new setting survives a power-cycle.
- No response was received after sending the command ‘r8#’. It seems that this register is not available all models of this sensor. Neither was a response received after an ‘s8-0#’ or an ‘s8-1#’ command.
- After sending an r-command, a line termination sequence is received, prior to the response on the command. The line termination sequence is “\r\r\n”. The line which is send as a response on the command also ends with the same line terminator sequence. There is however one exception: after command “r6#”, no line termination sequence is send.
- Some forum messages, as recent as august 2023, show an example program which sends a 16-bit binary number to retrieve the (real time?) distance. It has been verified that at least some models of this sensor do not respond after receiving such input.

2.4.4. I2C registers

The table below shows information about the registers within the sensor. The description, register address, length, permissible I/O mode and the range of values are shown. The result of a read command is a set of 1 or 2 bytes, the most significant one first. If a number can get negative, sign bit extension might be necessary.

Description	Address	Octets	Access	Unit	Range
Real time distance	0x00	2	Read only	mm	100 ~ 1800
Filtered distance	0x04	2	Read only	mm	100 ~ 1800
Update offset	0x06	2	Read/write	mm	-99 ~ 99. Writing 0 sets the offset to 0.
Distance mode	0x08	1	Read/write	-	0=filtered, 1=real time
Medium mode	0x09	1	Read/write	-	0=active send, 1=passive read
Maximum distance	0x0c	2	Read only	mm	100 ~ 1800
I2C address	0x0f	1	Read/write	-	0x02 ~ 0xfe, even (LSB = 0)

2.4.4.1. Various remarks

Using a Raspberry Pi (RPi) , running bookworm and using Python module smbus2 to access the sensor, it was found that:

- The sensor seems to perform some clock stretching when data is to be read from the sensor. It seems to hold the CLK-line low for an additional 10 [µs] when switching from reading to writing. The RPi does not react well, which is a well documented bug since 2013. It may cause the most significant bit of the byte following on the bus to become ‘1’ (at 40 [kHz]) or exceptions being thrown (at 100 [kHz]). With a bus frequency of 20 [kHz] the clock stretching is not visible any more and data can be exchanged without problems.
- Writing register 0x0c, the maximum distance, seems to succeed, but the new value is retained for only a very short time. After 25 [ms] the new value is forgotten, and the previous value is returned when (re)reading the register. So effectively, this register is a read-only register.