is now

indie Semiconductor FFO GmbH

To learn more about indie Semiconductor, please visit our website at
www.indiesemi.com

For customer support, please contact us at: dfo.support@indiesemi.com

Silicon Radar GmbH
Im Technologiepark 1
15236 Frankfurt (Oder)
Germany

fon +49.335.557 17 74
fax +49.335.557 10 50
http://www.siliconradar.com
support@siliconradar.com

Protocol Description SiRad Evaluation Kits

Status: Release	Date: 08 Jan. 19	Author: Silicon Radar GmbH	
Version: 2.1	Document number:	Filename: ProtocolDescription	Page: 1 of 32

Version Control

Version	Changed section	Description of change	Reason of change
1.0	all		Initial document
2.0	all	Content and appearance	Hardware & firmware update
2.1	all	Frame descriptions	Protocol update

Table of Contents

List of Figures

1 Measurement Flow

This section describes the measurement flow of the SiRad Evaluation Kit with its parameters. The measurement parameters as well as the kind and amount of transmitted data can be adapted using the SiRad Evaluation Kit communication protocol described in the following sections.

The signal flow of the radar measurement is shown in Figure 1. Each measurement cycle is initiated by either an internal 'Self-Trigger' (int) or an external 'Manual Trigger' (ext). Continuous measurements can be triggered with a certain trigger frequency (freq).

Once a trigger is received, the PLL is started that drives a frequency ramp from f_{Base} to (f_{Base} + BW) with the currently set base-frequency (f_{Base}) and the currently set bandwidth (BW). The radar frontend starts its detection in the chosen frequency range.

The AD converter (ADC) begins processing the chosen number of data samples (n_{Smp}) with a certain sample frequency (f_{Smp}). The received data is amplified either by a manually set gain value (once) or by a continuously recalculated automatically acquired gain value (cont.), further named Auto Gain Control (AGC) Mode. If set, a DC cancellation is done.

The current measurement is repeated n times (for a number of n_{Ramps} frequency ramps) and summed up internally. When downsampling is active (with a certain downsampling number n_{Down}), each n_{Down} ramps are averaged. The downsampled sums of the ramp measurements are then divided by the number of ramps n_{Ramps}.

The downsampled data is processed by a window function and then transformed by an FFT with n_{FFT} points. The magnitude, phase and other information is extracted from the FFT output for the target list. The targets are detected by the CFAR operator with its parameters CFsize, CFguard and CFthres that is applied to the FFT output. A target list is then created from the CFAR output and the data extracted from the FFT output.

All data is always transferred immediately after a measurement took place.

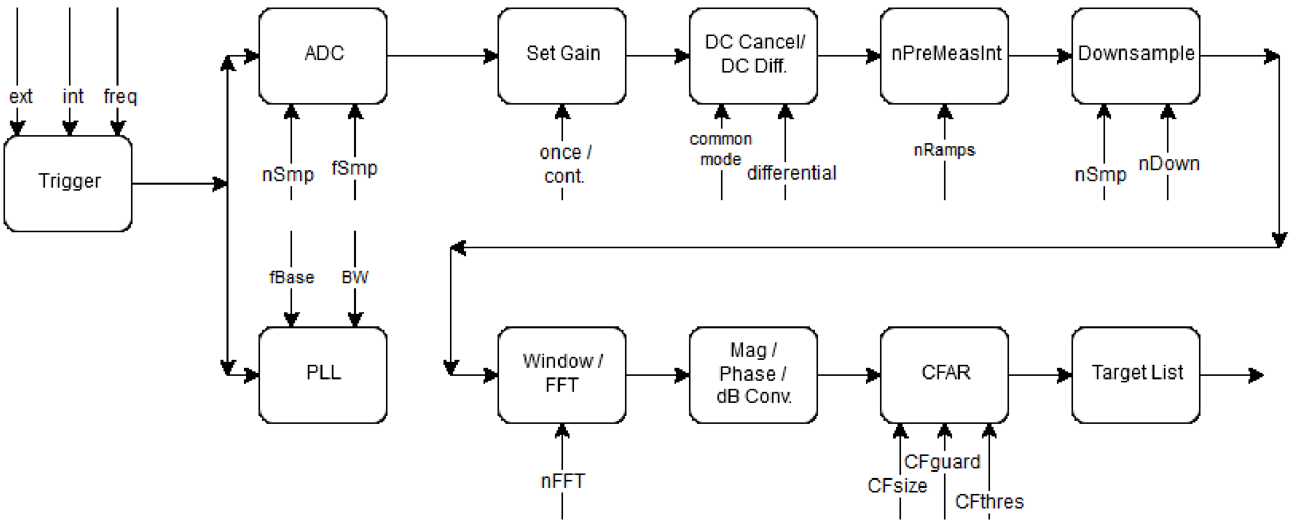

Figure 1: Measurement flow of the radar measurement on the SiRad Evaluation Kit

2 Standard Data

The SiRad Evaluation Kit communicates via UART. The UART protocol is (extended) ASCII based and includes standard data (explained in this section), commands (explained in Section 3) and extended data (explained in Section 4) – including raw data output.

The standard data communication is used by the Silicon Radar WebGUI. The command frames are only partly used by the Silicon Radar WebGUI. The extended data communication is not supported by the Silicon Radar WebGUI.

2.1 Data Blocks and Frame Formats

Once the SiRad Evaluation Kit is plugged in, it begins sending standard data. The standard data is transmitted in blocks of different data frames that are tied together in a single transmission, as highlighted in Figure 2. In the figure, two data blocks are marked red. Each data block ends with ASCII value 32 (' ', space) as stop marker and can contain multiple data frames of different size.

In the example in Figure 2, the data locks contain 5 data frames each. One data frame in the upper block is marked blue. Each data frame starts with ASCII value 33 ('!') as start marker and ends with two ASCII command characters ('CR' and 'LF') as stop marker.

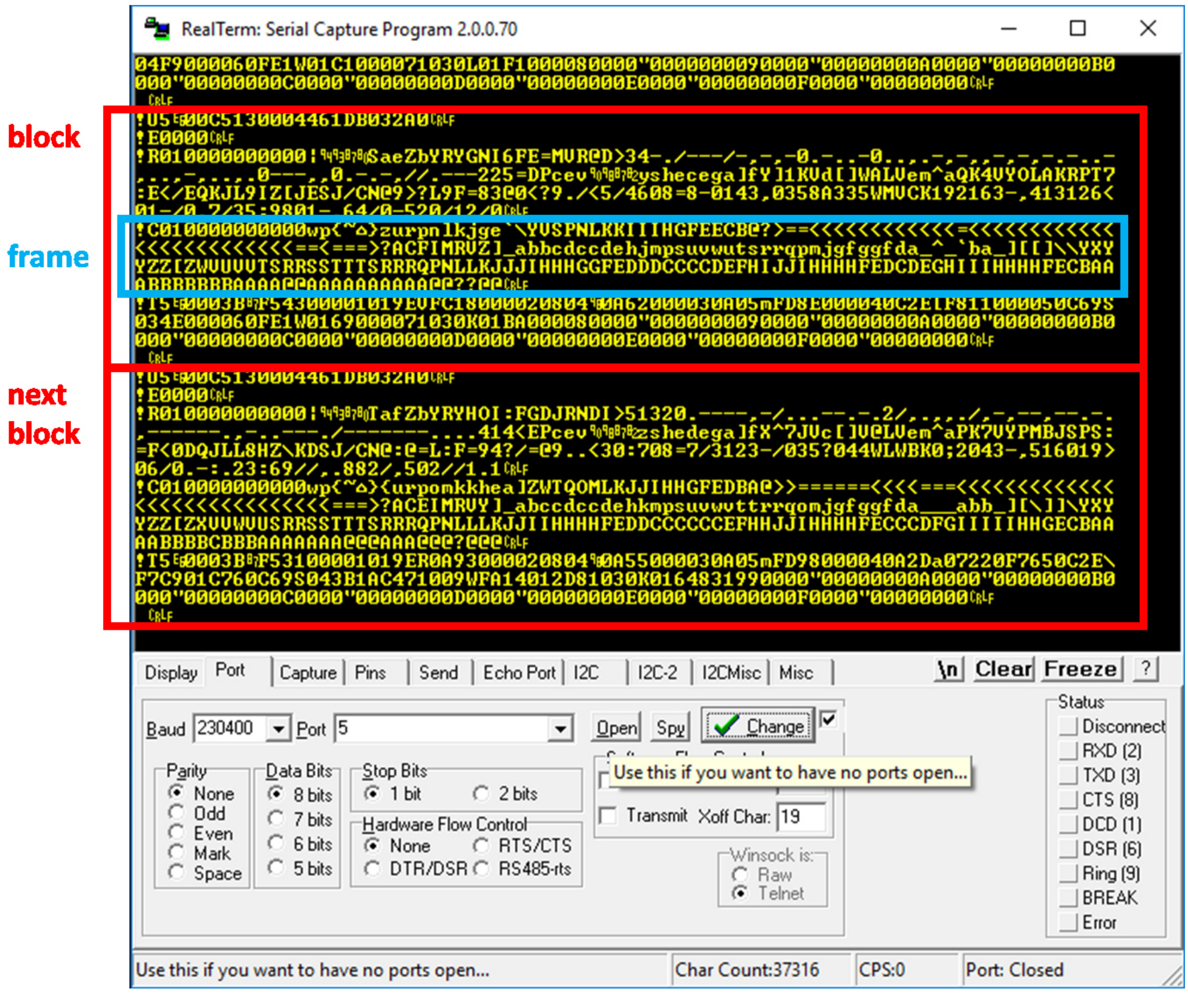

Figure 2: Standard data blocks and frames in a terminal window

Figure 3 shows the supported standard data frames and Table 1 lists their purpose. The blue parts in Figure 3 indicate start and stop markers and the frame identifier, orange and green parts indicate data parts and grey parts indicate reserved parts that should not be used.

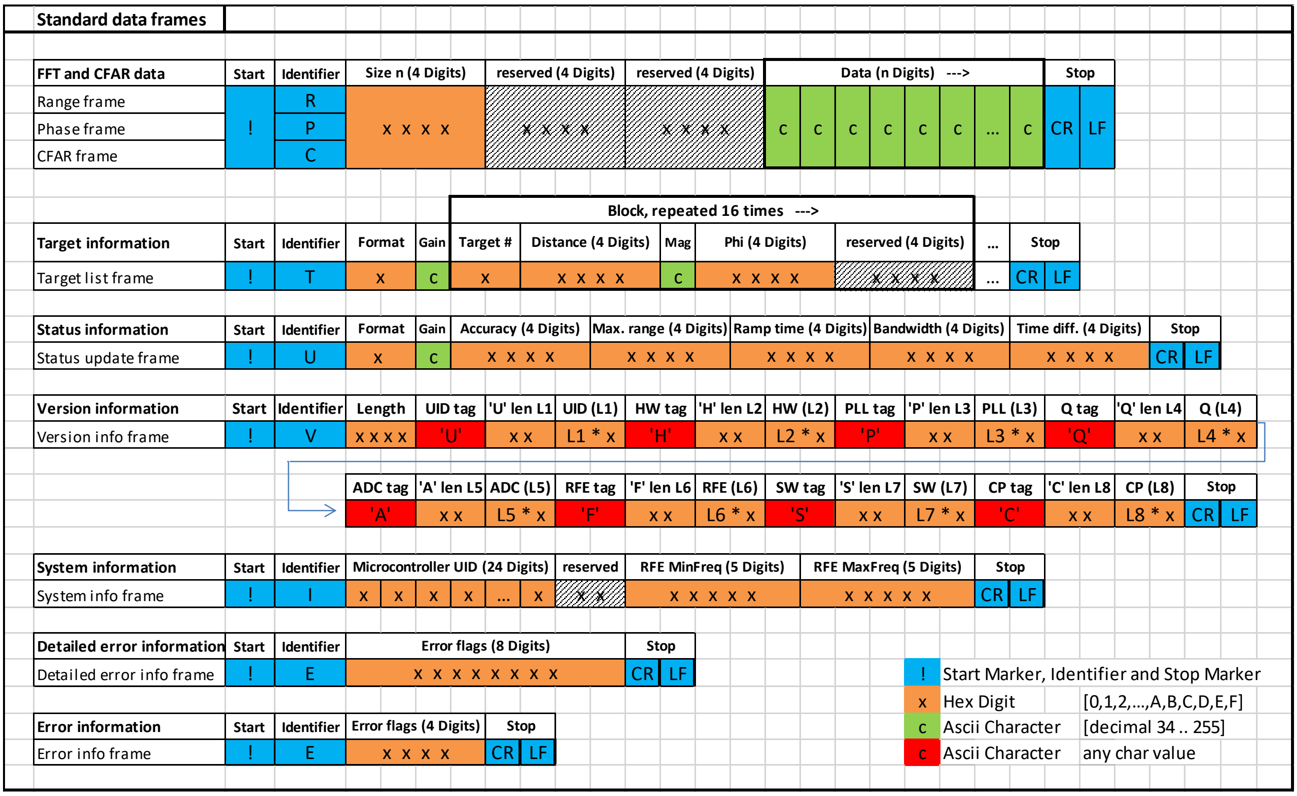

Figure 3: Standard data frames overview

There are several frame types that consist of multiple data values tied together to form a specific data packet, for example, a frame containing detected targets or a frame containing system information. Each frame type is recognized by a unique identifier (a certain letter) following the start marker of the frame, for example, the letter 'T' indicates a frame containing target data (the list of detected targets) or 'I' indicates a frame containing system information. The frame types are of different size. Please read the following sections about how to interpret the transmitted data in the frames.

Table 1: Standard data frames overview, description

Extended data frame	Description
Range frame	Contains distance data extracted from the FFT output
Phase frame	Contains phase information extracted from the FFT output
CFAR frame	Contains the output of the CFAR operator
Target list frame	Contains the target list with the detected targets
Version info frame	Contains hardware and firmware information
Status update frame	Contains status data updates of the SiRad Evaluation Kit
System info frame	Contains hardware information of the SiRad Evaluation Kit
Detailed error info frame	Contains detailed error information from the SiRad Evaluation Kit
Error info frame	Contains basic error information from the SiRad Evaluation Kit

2.2 FFT and CFAR data

The **range frame** contains the magnitude output of the FFT and the **phase frame** contains the argument of the FFT. The **CFAR frame** contains the output of the CFAR operator that is used to detect targets. The range frame, phase frame and CFAR frame share the same frame format, please see Figure 4. The start and stop markers and frame identifiers are highlighted in blue, data parts in orange and green color.

FFT and CFAR data	Start	Identifier	Size n (4 Digits)	reserved (4 Digits)	reserved (4 Digits)	Data (n Digits)								Stop	Stop
Range frame		R													
Phase frame	!	P	x x x x	x x x x	x x x x	c	c	c	c	c	c	...	c	CR	LF
CFAR frame		C													

Figure 4: Range, phase and CFAR frame format

After the frame's start marker (1 byte) and identifier 'R', 'P' or 'C' (1 byte) follows a 4 bytes 'Size' field, which indicates the number of data points (bytes) in the 'Data' field of the frame. There are two reserved fields of 4 bytes size between the 'Size' and the 'Data' field. The frame ends with the stop markers 'CR' + 'LF'.

Size field

The 'Size' field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with 4 'x' in Figure 4). The data is recognized as '0000' to 'FFFF' characters in the terminal output, and is interpreted as values between 0 and 65535, also see Table 2. For example, 'Size' = 0200 is interpreted as 0x0200, which is 512 in decimal range.

Please note, that this field is dependent on the chosen FFT size. However, a certain FFT size will lead to a half of the size of the FFT in the 'Size' field only. The FFT output is mirrored along the magnitude axis, so both parts are added together before the transmission and the length of the transmitted data is only half of the FFT output.

Table 2: Range, phase and CFAR frame - data encoding and interpretation

Field	Ecoding	Example	Interpretation	Allowed values
Size	x - 16 bit unsigned HEX between '0000' and 'FFFF'	'0200' -> 512	0 to 65535	'0010', '0020', '0040', '0080', '0100', '0200'
Data (range and CFAR frame)	c - characters between decimal value 34 and 254	letter 'Z' -> decimal 90	-140 to +80 dB in 220 steps	34 to 254
Data (phase frame)	c - characters between decimal value 34 and 254	letter 'Z' -> decimal 90	-π to +π rad (-180° to +180°) in 220 steps	34 to 254

Data field

The 'Data' field contains either the FFT output's magnitude (distance) data, the argument (phase) data or the CFAR output data, depending on the frame type.

The range and CFAR frame 'Data' bytes are transmitted as characters (marked with letters 'c' in Figure 4). The data is recognized as characters of decimal value 34 to 254 in the terminal output, and is interpreted as values between -140 and +80 dB in 220 steps of 1 dB, also see Table 2. For example, 'Data' = 'Z' is decimal 90 and means -84 dB.

The phase frame 'Data' bytes are transmitted as characters (marked with letters 'c' in Figure 4). The data is recognized as characters of decimal value 34 to 254 in the terminal output, and is interpreted as values between –π to +π in 220 steps, also see Table 2. For example, 'Data' = 'Z' is decimal 90 and means -1.54 rad (which is -88.36°).

2.3 Target Information

The target list contains the targets recognized by the CFAR operator. A target is detected whenever the magnitude of the FFT exceeds the CFAR operator's threshold. The local maximum of that area is marked as a target. The target list's frame format is shown in Figure 5.

								repeated 16x				
Target information	Start	Identifier	Format	Gain	Target #	Distance (4 Digits	Mag	Phi (4 Digits)	reserved (4 Digits)	...	Stop	Stop
Target list frame	!	T	x	c	x	x x x x	c	x x x x	x x x x	...	CR	LF

Figure 5: Target list frame format

The target list frame begins with the start marker (1 byte) and the identifier 'T' (1 byte) followed by the 'Format' field (1 byte), which indicates the unit format of the values in the 'Distance' field, and a 'Gain' value (1 byte) for the measurement. Then follows the target List with the target information, which is repeated 16 times (for 16 targets) and consists of a 'Target #' number (1 byte), the 'Distance' to the target (4 bytes), the 'Magnitude' or the signal strength of the target (1 byte), the argument or 'Phase' information of the target (4 bytes), and a reserved field (4 byte). The frame ends with the stop markers 'CR' + 'LF'.

Table 3: Target list frame - data encoding and interpretation of Format and Gain field

Field	Ecoding	Example	Interpretation	Allowed values
Format	x - unsigned HEX digit between '0' and 'F'	'F' -> 15	0 to 15	'5'
Gain	c - character between decimal value 34 and 254	letter 'Z' -> decimal 90	-140 to +80 dB in 220 steps	182, 195, 217, 230

<u>Format field</u>

The 'Format' field is encoded as an unsigned HEX digit in 1 transmitted byte (marked with an 'x' in Figure 5). The data is recognized as a '0' to 'F' character in the terminal output, and is interpreted as a value between 0 and 15, also see Table 3. For example, 'Format' = 5 is interpreted as 0x5, which is 5 in decimal range. Please see Table 4 for the meaning of the values in the 'Format' field.

Table 4: Target list frame - Format field

Format (HEX)	Description
0	relevant only for extended data
1	relevant only for extended data
2	relevant only for extended data
3	relevant only for extended data
4	reserved
5	distance in mm
6 to F	reserved

Table 5: Target list frame - Gain field

Gain (dec)	Description
8	8 dB gain
21	21 dB gain
43	43 dB gain
56	56 dB gain

<u>Gain field</u>

The 'Gain' field is transmitted as a character (marked with a 'c' in Figure 5). The data is recognized as a character of decimal value 34 to 254 in the terminal output, and is interpreted as a value between -140 and +80 dB in 220 steps of 1 dB, also see Table 3. For example, 'Gain' = 'Z' is decimal 90 and means -84 dB. There are currently four fixed gain settings available that depend on the hardware version, see Table 5.

Table 6: Target list frame - data encoding and interpretation of target list fields

Field	Ecoding	Example	Interpretation	Allowed values
Target #	x - unsigned HEX digit between ‘0’ and ‘F’	‘F’ -> 15	0 to 15	‘0’ to ‘F’
Distance	x - 16 bit unsigned HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	0 to 65535 in chosen unit	‘0000’ to ‘FFFF’
Magnitude	c - character between decimal value 34 and 254	letter ‘Z’ -> decimal 90	-140 to +80 dB in 220 steps	34 to 254
Phase	x - 16 bit signed HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	-32768 to +32767 (-π to +π rad)	-31416 to +31416

Target list

The target information is repeated 16 times in the target list. All 16 target information blocks are sent, regardless whether the target blocks are filled with detected targets or not. Empty target information blocks of the list are filled with zeros. Each target information block consists of the ‘Target #’, ‘Distance’, ‘Magnitude’, and ‘Phase’ fields.

Target field

The ‘Target #’ number field is encoded as an unsigned HEX digit in 1 transmitted byte (marked with an ‘x’ in Figure 5). The data is recognized as a ‘0’ to ‘F’ character in the terminal output, and is interpreted as a value between 0 and 15, also see Table 6. For example, ‘Target #’ = F is interpreted as 0xF, which is 15 in decimal range.

Distance field

The ‘Distance’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 5). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and is interpreted as values between 0 and 65535, also see Table 6. For example, ‘Distance’ = 0200 is interpreted as 0x0200, which is 512 in decimal range. The unit of the distance is determined by the value in the ‘Format’ field.

Magnitude field

The ‘Magnitude’ field is transmitted as a character (marked with a ‘c’ in Figure 5). The data is recognized as a character of decimal value 34 to 254 in the terminal output, and is interpreted as a value between -140 and +80 dB in 220 steps of 1 dB, also see Table 6. For example, ‘Magnitude’ = ’Z’ is decimal 90 and means -84 dB.

Phase field

The ‘Phase’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 5). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and is interpreted as values between -32768…+32767, also see Table 6.

2.4 Status Information

The status update frame in Figure 6 is a feedback of the current accuracy, range, ramp time, and ramp bandwidth and also returns the time since the last measurement.

Status information	Start	Identifier	Format	Gain	Accuracy (4 Digits)	Max. range (4 Digits)	Ramp time (4 Digits)	Bandwidth (4 Digits)	Time diff. (4 Digits)	Stop	Stop
Status update frame	!	U	x	c	x x x x	x x x x	x x x x	x x x x	x x x x	CR	LF

Figure 6: Status update frame format

The status update frame begins with the start marker (1 byte) and the identifier ‘U’ (1 byte) followed by the ‘Format’ field (1 byte), which indicates the unit format of the values in the ‘Max. Range’ field, and a ‘Gain’ value (1 byte) for the last measurement. Then follows the ‘Accuracy’ field (4 byte) currently configured settings, the ‘Max Range’ field (4 bytes) with the maximum detectable range with the current settings, the currently used ‘Ramp time’ (4 byte), the ‘Bandwidth’ field (4 bytes), and the time passed since the last measurement in the ‘Time diff.’ field (4 byte). The frame ends with the stop markers ‘CR’ + ‘LF’.

Table 7: Status update frame - data encoding and interpretation of Format, Gain, and Accuracy fields

Field	Ecoding	Example	Interpretation	Allowed values
Format	x - unsigned HEX digit between ‘0’ and ‘F’	‘F’ -> 15	0 to 15	‘5’
Gain	c - character between decimal value 34 and 254	letter ‘Z’ -> decimal 90	-140 to +80 dB in 220 steps	182, 195, 217, 230
Accuracy	x - 16 bit unsigned HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	0 to 65535 (0 to 6553.5 mm)	‘0000’ to ‘FFFF’

<u>Format field</u>

The ‘Format’ field is encoded as an unsigned HEX digit in 1 transmitted byte (marked with an ‘x’ in Figure 6). The data is recognized as a ‘0’ to ‘F’ character in the terminal output, and is interpreted as a value between 0 and 15, also see Table 7. For example, ‘Format’ = 5 is interpreted as 0x5, which is 5 in decimal range. Please see Table 4 for the meaning of the values in the ‘Format’ field.

<u>Gain field</u>

The ‘Gain’ field is transmitted as a character (marked with a ‘c’ in Figure 6). The data is recognized as a character of decimal value 34 to 254 in the terminal output, and is interpreted as a value between -140 and +80 dB in 220 steps of 1 dB, also see Table 7. For example, ‘Gain’ = ’Z’ is decimal 90 and means -84 dB. There are currently four fixed gain settings available that depend on the hardware version, see Table 5.

<u>Accuracy field</u>

The ‘Accuracy’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 6). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and are interpreted as values between 0 to 65535, which are translated to an accuracy of 0 to 6553.5 mm with 0.1 mm resolution, also see Table 7. For example, ‘Accuracy’ = 0200 is interpreted as 0x0200, which is 512 in decimal range and translates to an accuracy of 51.2 mm.

Table 8: Status update frame - data encoding and interpretation of Max. range, Ramp time, Bandwidth, and Time diff. fields

Field	Ecoding	Example	Interpretation	Allowed values
Max. Range	x - 16 bit unsigned HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	0 to 65535 in chosen unit	‘0000’ to ‘FFFF’
Ramp time	x - 16 bit unsigned HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	0 to 65535 in us	‘0000’ to ‘FFFF’
Bandwidth	x - 16 bit unsigned HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	0 to 65535 in MHz	‘0000’ to ‘FFFF’
Time diff.	x - 16 bit unsigned HEX between ‘0000’ and ‘FFFF’	‘0200’ -> 512	0 to 65535 (0 to 0.65535 s)	‘0000’ to ‘FFFF’

Max. Range field

The ‘Max. Range’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 6). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and are interpreted as values between 0 and 65535 in the chosen distance unit, also see Table 8. For example, ‘Max. Range’ = 0200 is interpreted as 0x0200, which is 512 in decimal range. The unit of the distance is determined by the value in the ‘Format’ field.

Ramp time field

The ‘Ramp time’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 6). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and are interpreted as values between 0 to 65535 in us, also see Table 8 For example, ‘Ramp time’ = 0200 is interpreted as 0x0200, which is 512 in decimal range.

Bandwidth field

The ‘Bandwidth’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 6). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and are interpreted as values between 0 and 65535 in MHz, also see Table 8. For example, ‘Bandwidth’ = 0200 is interpreted as 0x0200, which is 512 in decimal range.

Time diff. field

The ‘Time diff.’ field is encoded as a 16 bit unsigned HEX number in 4 transmitted bytes (marked with ‘x’ in Figure 6). The data is recognized as ‘0000’ to ‘FFFF’ characters in the terminal output, and are interpreted as values between 0 to 65535, which translates to 0 to 0.65535 seconds in 10 ms steps, also see Table 8. For example, ‘Time diff.’ = 0200 is interpreted as 0x0200, which is 512 in decimal range. The counter runs at 100 kHz and is configured as an overflowing 16-bit counter. Each tick lasts 10 ms and the counter overflows at 0.65535 seconds. Therefore, the minimum unambiguous measurement frequency is 1.5 Hz.

2.5 Version Information

The version frame is used to uniquely identify the SiRad Evaluation Kit and returns information about the hardware and firmware, see Figure 7.

Version information	Start	Identifier	Length	UID tag	'U' len L1	UID (L1)	HW tag	'H' len L2	HW (L2)	PLL tag	'P' len L3	PLL (L3)	Q tag	'Q' len L4	Q (L4)
Version info frame	!	V	x x x x	'U'	x x	L1 * x	'H'	x x	L2 * x	'P'	x x	L3 * x	'Q'	x x	L4 * x
			ADC tag	'A' len L5	ADC (L5)	RFE tag	'F' len L6	RFE (L6)	SW tag	'S' len L7	SW (L7)	CP tag	'C' len L8	CP (L8)	Stop
			'A'	x x	L5 * x	'F'	x x	L6 * x	'S'	x x	L7 * x	'C'	x x	L8 * x	CR LF

Figure 7: Version information frame format

Length field

Contains the length of the version frame excluding the start marker, identifier, the length field itself and the stop markers. Field size: 4 hex chars.

UID tag (‘U’)

Indicates the start of the UID field. Size: 1 hex char.

‘U’ len L1,

Contains the length of the UID field (number of chars). Field size: 2 hex chars.

UID (L1) field
The 'Microcontroller UID' field is a unique 24 byte unsigned HEX number, also see Table 9. Field size: variable.

HW tag ('H')
Indicates the start of the HW field. Size: 1 hex char.

'H' len L2
Contains the length of the HW field (number of chars). Field size: 2 hex chars.

HW (L2) field
Contains the baseboard hardware identifier, for example, 'EA' for the SiRad Easy or 'SI' for the SiRad Simple. Field size: variable.

PLL tag ('P')
Indicates the start of the PLL field. Size: 1 hex char.

'P' len L3
Contains the length of the PLL field (number of chars). Field size: 2 hex chars.

PLL (L3) field
Contains the PLL chip identifier, for example, '59' for the ADF4159. Field size: variable.

CLK tag ('Q')
Indicates the start of the CLK field. Size: 1 hex char.

'Q' len L4
Contains the length of the CLK field (number of chars). Field size: 2 hex chars.

CLK (L4) field
Contains the CLK chip identifier. Field size: variable.

ADC tag ('A')
Indicates the start of the ADC field. Size: 1 hex char.

'A' len L5
Contains the length of the ADC field (number of chars). Field size: 2 hex chars.

ADC (L5) field
Contains the operating mode of the ADC, for example, 'I' for interleaved mode or 'N' non-interleaved mode. Field size: variable.

RFE tag ('F')
Indicates the start of the RFE field. Size: 1 hex char.

‘F’ len L6

Contains the length of the RFE field (number of chars). Field size: 2 hex chars.

RFE (L6) field

Contains the radar front end chip identifier, for example, ‘120_0x’ for the 120 GHz chip, ‘024_0x’ for the 24 GHz chip, or ‘UIDENT’ if the radar chip was not identified. Field size: variable.

Software version tag (‘S’)

Indicates the start of the software version field. Size: 1 hex char.

‘S’ len L7

Contains the length of the software version field (number of chars). Field size: 2 hex chars.

Software version (L7) field

Contains the software version as described below. Field size: variable.

<check-in ID >-<date >-<major>.<minor>.<revision>

Communication protocol version tag (‘C’)

Indicates the start of the protocol version field. Size: 1 hex char.

‘C’ len L8

Contains the length of the protocol version field (number of chars). Field size: 2 hex chars.

Protocol version (L8) field

Contains the protocol version as described below. Field size: variable.

<protocol ID>-<spec date>-<major>.<minor>.<revision>

2.6 System Information

The system info frame is used to uniquely identify different SiRad Evaluation Kits and return radar frontend information, see Figure 8.

System information	Start	Identifier	Microcontroller UID (24 Digits)						reserved	RFE MinFreq (5 Digits)	RFE MaxFreq (5 Digits)	Stop	Stop
System info frame	!	I	x	x	x	x	...	x	x x	x x x x x	x x x x x	CR	LF

Figure 8: System information frame format

After the start marker (1 byte) and the identifier ‘I’ (1 byte) follows the ‘UID’ field (24 bytes), which carries the UID of the microcontroller on the SiRad Evaluation Kit. Afterwards, there is a 2 byte reserved field then follow two 5 byte fields, which contain the minimum and maximum frequencies of the SiRad Evaluation Kit’s radar frontend. The frame ends with the stop markers ‘CR’ + ‘LF’.

Table 9: Target list frame - data encoding and interpretation of target list fields

Field	Ecoding	Example	Interpretation	Allowed values
Microcontroller UID	x - 24 byte string	'800F0011570A 463332322039'	-	-
RFE MinFreq	x - 16 bit unsigned HEX between '00000' and 'FFFFF'	'1D0D8' -> 119000	0 to 119000 in MHz	'00000' to 'FFFFF'
RFE MaxFreq	x - 16 bit unsigned HEX between '00000' and 'FFFFF'	'1E848' -> 125000	0 to 125000 in MHz	'00000' to 'FFFFF'

Microcontroller UID field

The 'Microcontroller UID' field is a unique 24 byte unsigned HEX number (marked with 'x' in Figure 8), also see Table 9.

RFE MinFreq field

The 'RFE MinFreq' field is encoded as a 20 bit unsigned HEX number in 5 transmitted bytes (marked with 'x' in Figure 8). The data is recognized as '00000' to 'FFFFF' characters in the terminal output, and are interpreted as values between 0 and 1048575 in MHz, also see Table 9. For example, 'RFE MinFreq' = 1D0D8 is interpreted as 0x1D0D8, which is 119000 in decimal range.

RFE MaxFreq field

The 'RFE MaxFreq' field is encoded as a 20 bit unsigned HEX number in 5 transmitted bytes (marked with 'x' in Figure 8). The data is recognized as '00000' to 'FFFFF' characters in the terminal output, and are interpreted as values between 0 and 1048575 in MHz, also see Table 9. For example, 'RFE MaxFreq' = 1E848 is interpreted as 0x1E848, which is 125000 in decimal range.

2.7 Detailed Error Information

The detailed error info frame includes error bits that may be raised during the signal processing of the radar data, see Figure 9: Detailed error information frame format. This frame can be requested by sending the !E command and will be send in addition to the basic error information frame below.

Detailed error information	Start	Identifier	Error flags (8 Digits)	Stop	
Detailed error info frame	!	E	x x x x x x x x	CR	LF

Figure 9: Detailed error information frame format

The detailed error info frame begins with the start marker (1 byte) and the identifier 'E' (1 byte) followed by the 'Error flags' field (8 byte), which is zero when no errors have been detected. The frame ends with the stop markers 'CR' + 'LF'.

Error flags field

The 'Error flags' field is transmitted as a 8 byte unsigned HEX number (marked with 'x' in Figure 9). Figure 10 shows the error bits in the 'Error flags' field. The error bits are explained below.

	Data								Processing								Basisband								Frontend							
Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ERROR_DETAILED	CRC	CRC	CRC	CRC	FLS	FLS	FLS	FLS	FFT	FFT	FFT	FFT	ADC	ADC	ADC	ADC	AMP	AMP	AMP	AMP	PLL	PLL	PLL	PLL	RFE	RFE	RFE	RFE	RFE	RFE	RFE	RFE

CRC	0	no error	FFT	0	no error	AMP	0	no error	RFE	0	no error
	1	reserved		1	reserved		1	reserved		1	reserved
CRC	0	no error	FFT	0	no error	AMP	0	no error	RFE	0	no error
	1	reserved		1	reserved		1	reserved		1	reserved
CRC	0	no error	FFT	0	no error	AMP	0	no error	RFE	0	no error
	1	reserved		1	reserved		1	reserved		1	RFE out of spec
CRC	0	no error	FFT	0	no error	AMP	0	no error	RFE	0	no error
	1	reserved		1	reserved		1	Saturation		1	VCO error
FLS	0	no error	ADC	0	no error	PLL	0	no error	RFE	0	no error
	1	reserved		1	reserved		1	reserved		1	BW overrun
FLS	0	no error	ADC	0	no error	PLL	0	no error	RFE	0	no error
	1	reserved		1	reserved		1	Lock loss		1	BW underrun
FLS	0	no error	ADC	0	no error	PLL	0	no error	RFE	0	no error
	1	reserved		1	DC error		1	Fmax not found		1	Fbase high
FLS	0	no error	ADC	0	no error	PLL	0	no error	RFE	0	no error
	1	reserved		1	Sample overrun		1	Fmin not found		1	Fbase low

Figure 10: Detailed error flags

Error domains:

- CRC: <reserved>
- FLS: <reserved>
- FFT: <reserved>
- ADC: temporary ADC, sampling and data buffering errors
- AMP: temporary amplification errors, for example, saturation
- PLL: temporary PLL configuration errors, for example, operating range exceeded
- RFE: temporary radar frontend configuration errors, for example, operating range exceeded

Temporary errors are raised during processing but may go away when the parameter setting is changed. For example, when the parameters for the front end are manually changed so that its operating range is exceeded, a temporary RFE or PLL error may appear as long as this setting is applied.

2.8 Error Information

The error info frame includes error bits that may be raised during the signal processing of the radar data, see Figure 11. This frame will be send by default when the status update is enabled.

Error information	Start	Identifier	Error flags (4 Digits)	Stop	Stop
Error info frame	!	E	x x x x	CR	LF

Figure 11: Error information frame format

The error info frame begins with the start marker (1 byte) and the identifier 'E' (1 byte) followed by the 'Error flags' field (4 byte), which is zero when no errors have been detected. The frame ends with the stop markers 'CR' + 'LF'.

Error flags field

The 'Error flags' field is transmitted as a 4 byte unsigned HEX number (marked with 'x' in Figure 11). Figure 12 shows the error bits in the 'Error flags' field. The error bits are explained below.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ERROR	reserved								reserved								reserved		FLS	PRC	BB	PLL	RFE	CRC	reserved							

FLS	
0	no error
1	Flash error

BB	
0	no error
1	Baseband error

RFE	
0	no error
1	Frontend error

PRC	
0	no error
1	Processing error

PLL	
0	no error
1	PLL error

CRC	
0	no error
1	CRC error

Figure 12: Error flags

Error domains:

- FLS: <reserved>
- PRC: temporary errors in the signal processing
- BB: temporary baseband processing errors
- PLL: temporary PLL configuration errors
- RFE: temporary radar frontend configuration errors
- CRC: temporary errors in the UART transmission or CRC checksum

Temporary errors are raised during processing but may go away when the parameter setting is changed. For example, when the parameters for the front end are manually changed so that its operating range is exceeded, a temporary RFE or PLL error may appear as long as this setting is applied.

3 Commands

3.1 Frame Formats

The command frames are only partly used by the Silicon Radar WebGUI. Each command frame starts with ASCII value 33 ('!') as start marker and ends with two ASCII command characters ('CR' and 'LF') as stop marker.

Command frames

Configuration command	Start	Identifier	Command settings (8 Digits)	Stop	
System configuration	!	S	SYS_CONFIG	CR	LF
Radar frontend configurati		F	RFE_CONFIG		
PLL configuration		P	PLL_CONFIG		
Baseband setup		B	BB_CONFIG		
Programming mode		W	ProgMode (fixed)		

Special function command	Start	Identifier	Stop	
Get full error report	!	E	CR	LF
Get system info		I		
Do frequency scan		J		
Set to max. bandwidth		K		
Configure Pre-Trigger		L		
Trigger		M		
Complete cycle (L,M)		N		
Get version info		V		

!	Start Marker, Identifier and Stop Marker	
x	Hex Digit	[0,1,2,...,A,B,C,D,E,F]
c	Ascii Character	[decimal 34 .. 255]

Figure 13: Command frames

The command frames are used to transmit configuration data to the SiRad Evaluation Kit. Different commands are used to configure several functionalities. Figure 13 shows the available command frames and Table 10 lists their purpose. The blue parts in Figure 13 indicate start and stop markers, orange parts indicate data parts. The commands are further explained in the subsections below.

Table 10: Command frames overview, description

Command frame	Description
System configuration	Configure basic functions of the system
Radar frontend configuration	Configure frontend base-frequency and the VCO Divider
PLL configuration	Configure the bandwidth of the frequency ramp
Baseband setup	Configure baseband and processing related parameters
Get full error info	Request detailed error data
Get system info	Request system info data
Do frequency scan	Request a scan of the max usable bandwidth of the installed frontend
Set to max. bandwidth	Set ramp bandwidth to the measured maximum
Send Pre-Trigger	Send a Pre-trigger for an automatic gain measurement (AGC Mode)
Send Trigger	Send a trigger for a measurement
Send Pre-Trigger + Trigger (L, M)	Send Pre-trigger and trigger in one command
Get version info	Request version info data

3.2 Send Commands

You can use a terminal program to send the command strings as, for example, shown in Figure 16 with the Realterm terminal program. Calculate the command string by converting the command bits, an example is shown in Figure 14, into hex format, shown in Figure 15. Use zeros for any RESERVED fields.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay				reserved		LED		reserved							RAW	res	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
Binary	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1	0

Figure 14: Example SYS_CONFIG command bits

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay				reserved		LED		reserved							RAW	res	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
HEX	0				0				0				0				4				9				B				A			

Figure 15: Example SYS_CONFIG command in hex format

Add the start marker '!' and the frame identifier (S, F, P, B) to the front of the hex command to form the command string. In case of the special function commands just use the start marker '!' and the command identifier (I, J, K, L, M, N) as the command string. In case of the example in Figure 15, you would get the command string

!S000049BA

In case of a special function command, it could look like this

!M

Paste the command string into one of the send fields like shown in Figure 16, activate CR and LF to let Realterm add the stop markers to the string automatically, and then click 'Send ASCII'.

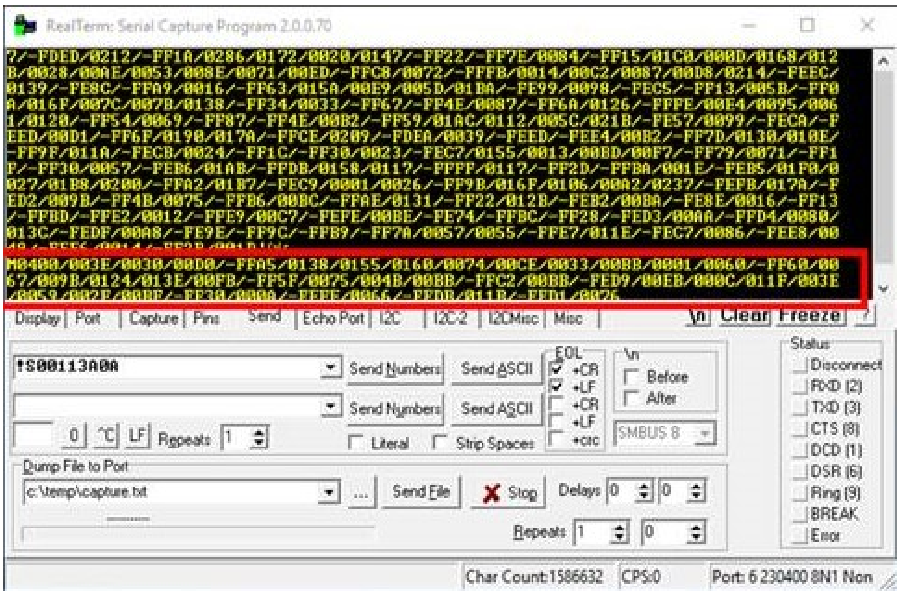

Figure 16: Send commands using the Realterm terminal program

3.3 System Configuration

The system configuration command in Figure 17 is used to configure basic functions of the system, such as the triggering, frontend type, LED, data output interface, gain, DC cancellation and data output modes.

	Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG		SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE

SelfTrigDelay				
0	0	0	2	ms
0	0	1	4	ms
0	1	0	8	ms
0	1	1	16	ms
1	0	0	32	ms
1	0	1	64	ms
1	1	0	128	ms
1	1	1	256	ms

LED		
0	0	off
0	1	1st target rainbow
1	0	reserved
1	1	reserved

RAW	
0	raw ADC output off
1	raw ADC output on

AGC	
0	auto gain control off
1	auto gain control on

Gain			
0	0	8	dB
0	1	21	dB
1	0	43	dB
1	1	56	dB

SER2	
0	output on SER2 off
1	output on SER2 on

SER1	
0	output on SER1 off
1	output on SER1 on

EXT	
0	standard sensor data
1	extended sensor data

ST	
0	status frames off
1	status frames on

TL	
0	target list frames off
1	target list frames on

P	
0	phase frames off
1	phase frames on

C	
0	CFAR frames off
1	CFAR frames on

DC	
0	DC cancellation off
1	DC cancellation on

SLF	
0	manual (ext) trigger
1	self-trigger mode

PRE	
0	pre-trigger off
1	pre-trigger on

R	
0	distance frames off
1	distance frames on

Figure 17: System configuration, SYS_CONFIG command frame

LED (2 bit) - LED operation (Supported by WebGUI)

Selects the behavior of the onboard LED. The LED is switched off when disabled. When enabled to 1st target rainbow, the LED displays the distance of the first recognized target as a color from blue (far) over green (medium range) to red (close). The current maximum range is used as a reference.

AGC (1 bit) - Auto Gain Control (Supported by WebGUI)

Activates the Auto Gain Control (AGC) Mode when enabled. Overrides the manual settings in the 'Gain' field. The AGC Mode uses 2 ramps at the beginning of the measurement phase or the pre-trigger phase for the gain measurement (depending on whether 'Pre-trigger' is switched on). Also see the next section about the triggering options.

Gain (2 bit) - Manual Gain Control (Supported by WebGUI)

Select the gain manually between 4 modes. This bit is overridden by the AGC bit, which enables the Auto Gain Control (AGC) Mode.

SER1 / SER2 (1 bit each) - Output Interface (Supported by WebGUI)

Use these bits to choose the UART output interface of the SiRad Evaluation Kit. Configuration data can be fed to the SiRad Evaluation Kit using both UARTs at any time. Select SER1 for the WiFi connection of the SiRad Easy® and SER2 for the USB connection of the SiRad Easy®. Select SER1 for any output interface (UART-USB or WiFi) of the SiRad Simple®.

DC (1 bit) - DC Cancellation (Supported by WebGUI)

Activates de-trending and static offset compensation in the digital domain.

ST, TL, P, C, R (1 bit, each) - Data Frames (Supported by WebGUI)

Enables or disables the status update frame (ST), target list frame (TL), phase frame (P), CFAR frame (C), and/or the range frame (R). When these bits are enabled, the according frame will be output after each measurement. Use these bits to switch the transmission of these frames on or off. The transmission of the frames can also be switched on or off separately in the Silicon Radar WebGUI. **Switching unnecessary frames off can increase the update rate of the SiRad Evaluation Kit significantly.**

3.3.1 Trigger Options

The measurement is divided into two parts: pre-measurement and measurement. The pre-phase is used to figure out the optimal gain setting of the system without saturation. It uses two frequency ramps to do that. After the pre-measurement phase the actual measurement is started and consists of a pre-defined number of frequency ramps. The measurement can be triggered either manually (externally) via the external trigger line, a command frame or internally via a timer (self-trigger).

PRE (1 bit) - Pre-trigger (Supported by WebGUI)

When using the external trigger options, a pre-trigger can be used to measure saturation and do the gain settings. When this bit is enabled, the system waits for 10 ms for the main trigger. If the main trigger does not occur within this time, the system will go back to sleep mode. The pre-trigger option can be useful to synchronize a number of SiRad Evaluation Kits and start their measurements simultaneously or at a defined time.

SLF (1 bit) - Manual trigger or Self-trigger (Supported by WebGUI)

When this bit is disabled, the system enters deep sleep mode after transmitting data and waits for an external trigger (Manual Trigger Mode). This is useful to minimize power consumption of the system when using longer measurement intervals.

When this bit is enabled, the SiRad Evaluation Kit triggers each measurement with an internal timer after 100 ms (Self-Trigger Mode). ‘Pre-trigger’ and ‘Manual Trigger’ are overridden with this bit.

SelfTrigDelay (3 bit) – Self-Trigger Delay (Supported by WebGUI)

Sets a delay time between self-trigger events.

3.3.2 RAW Data Options

The Extended Data Mode has to be activated to enable raw data output. This is done by setting the EXT bit in the SYS_CONFIG command frame. Further, you have to choose the kind of raw data you want to transmit by enabling or disabling the RAW bit and by setting the desired output format in the ‘Format’ field of the BB_CONFIG frame.

EXT (1 bit) - Extended Data (Supported by WebGUI)

When enabled, this bit activates the Extended Data Mode for the FFT data, CFAR data and target information, and also enables raw data output. In the Extended Data Mode, signed and unsigned data is transmitted in 16-bit HEX numbers in linear scale. The extended data frames are described in Section 4. The extended data communication is not supported by the Silicon Radar WebGUI.

RAW (1 bit)

When enabled, pure raw data is transmitted from the ADC without any preprocessing. When disabled, the raw data is processed by a window function before transmission. You have to set the RAW bit together with the according setting in the ‘Format’ field of the BB_CONFIG command frame.

If the RAW bit is enabled, you also have to set the ‘mm’ option in the ‘Format’ field of the BB_CONFIG command frame.

If the RAW bit is disabled, you also have to set the raw A/D option in the ‘Format’ field of the BB_CONFIG command frame.

The raw data is transmitted after a complete measurement cycle. If more than one ramp is captured, they are summed up first and then transmitted.

3.3.3 Default Settings

Below you can find the default settings of the WebGUI for the SYS_CONFIG command frame to speed up your development by simple copy and paste into your favorite terminal program.

WebGUI Default Setting (SiRad Easy® with 24 GHz frontend):

!S010049BA	No self-trigger delay, LED on, RAW data off, AGC Mode on, SER2 (USB) output on, SER1 (WiFi) output off, EXT Mode off, status update data on, target list data on, CFAR data on, range data on, DC cancellation on, Self-Trigger Mode, Pre-Trigger off

WebGUI Default Setting (SiRad Easy® with 122 GHz frontend):

!S000049BA / !S001049BA*	No self-trigger delay, LED off, RAW data off, AGC Mode on, SER2 (USB) output on, SER1 (WiFi) output off, EXT Mode off, status update data on, target list data on, CFAR data on, range data on, DC cancellation on, Self-Trigger Mode, Pre-Trigger off

WebGUI Default Setting (SiRad Simple®):

!S000045BA / !S001045BA*	No self-trigger delay, LED off, RAW data off, AGC Mode on, SER2 output off, SER1 (UART-USB or WiFi) output on, EXT Mode off, status update data on, target list data on, CFAR data on, range data on, DC cancellation on, Self-Trigger Mode, Pre-Trigger off

* Obsolete default configuration word. The third digit is a reserved field and can be ignored (or set to zero).

3.4 Radar Frontend Configuration

The radar frontend configuration command in Figure 18 is used to configure the radar frontend's base-frequency and the VCO Divider.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
RFE_CONFIG	VCO Divider (13 Bits, fixed per frontend)													Radar Frontend Base Frequency [MHz] (19 Bits)																		

VCO Divider (13 Bits, fixed per frontend)													
0	0	0	0	0	0	0	0	0	1	0	0	0	24 GHz
0	0	0	0	0	0	1	0	0	0	0	0	0	122 GHz
x	x	x	x	x	x	x	x	x	x	x	x	x	reserved

Radar Frontend Base Frequency [MHz] (19 Bits)																			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 MHz
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1 MHz
…																			…
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	524287 MHz

Figure 18: Radar frontend configuration, RFE_CONFIG command frame

<u>VCO Divider (13 bit)</u>

The VCO divider is a 13-bit unsigned integer value, so the theoretic value range is 0 to 8191. Please note, that the VCO divider is fixed in hardware and frontend specific. The values for the 24 GHz and 122 GHz frontends are given in Figure 18. The SiRad Simple® has a fixed 122 GHz frontend onboard.

<u>RF Base-Frequency (19 bit)</u>

The base-frequency is a 19-bit unsigned integer value interpreted in MHz, so the theoretic value range is 0 to 524287 MHz. Please note, that each frontend has a slightly different minimum and maximum operating frequency due to production tolerances. The frequencies supported by your frontend should be approximately in the range of 23300 to 26200 MHz for the 24 GHz frontend and 119100 to 125900 MHz for the 122 GHz frontend. The base-frequency should be chosen at least 100 MHz above the minimum operating frequency. The base-frequency plus the chosen bandwidth should not exceed the maximum operating frequency minus 100 MHz for an unsaturated signal.

3.4.1 Default Settings

Below you can find the default settings of the WebGUI for the RFE_CONFIG command frame to speed up your development by simple copy and paste into your favorite terminal program.

WebGUI Default Setting (SiRad Easy® with 24 GHz frontend):

!F00405A3C	VCO divider = 8, RF base-frequency = 23100MHz

WebGUI Default Setting (SiRad Simple® and Easy® with 122 GHz frontend):

!F0201DC90	VCO divider = 64, RF base-frequency = 122000MHz

3.5 PLL Configuration

The PLL configuration command in Figure 19 is used to configure the bandwidth of the frequency ramp.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
PLL_CONFIG	reserved																Bandwidth [MHz] (16 Bits)															

Bandwidth [MHz] (16 Bits)																
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-32768 MHz
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-32767 MHz
…																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 MHz
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1 MHz
…																…
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	+32767 MHz

Figure 19: PLL configuration, PLL_CONFIG command frame

Bandwidth (16 bit)

The bandwidth word is a signed 16-bit value interpreted in MHz, so the theoretic value range is -32768 to +32767 MHz. Negative values result in a falling ramp slope, positive value provide a rising sawtooth shape. Please note that the maximum bandwidth supported by the 24 GHz frontend is around 2900 MHz and by the 122 GHz frontend is around 6800 MHz.

3.5.1 Default Settings

Below you can find the default settings of the WebGUI for the PLL_CONFIG command frame to speed up your development by simple copy and paste into your favorite terminal program.

WebGUI Default Setting (SiRad Easy® with 24 GHz frontend):

!P000003E8	Bandwidth = 1000 MHz

WebGUI Default Setting (SiRad Simple® and Easy® with 122 GHz frontend):

!P00001388	Bandwidth = 5000 MHz

3.6 Baseband Setup

The baseband configuration command in Figure 20 is used to configure baseband and processing related parameters, like data output options, the CFAR parameters, FFT parameters and sampling parameters.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		

CFAR Threshold [dB]					dB
0	0	0	0	0	0
0	0	0	0	1	1
…					…
1	1	1	1	1	31

Format			
0	0	0	raw A/D
0	0	1	FFT comp
0	1	0	FFT mag/ph
0	1	1	reserved
1	0	0	reserved
1	0	1	dist [mm]
1	1	0	reserved
1	1	1	reserved

FFT Size			
0	0	0	32
0	0	1	64
0	1	0	128
0	1	1	256
1	0	0	512
1	0	1	1024
1	1	0	reserved
1	1	1	reserved

CFAR Size				
0	0	0	0	0
0	0	0	1	1
…				
1	1	1	1	15

Downsampling			
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	4
1	0	0	8
1	0	1	16
1	1	0	32
1	1	1	64

CFAR Grd		
0	0	0
0	1	1
1	0	2
1	1	3

# Ramps			
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Average n			
0	0	0	0
0	0	1	1
…			
1	1	1	7

# Samples			
0	0	0	32
0	0	1	64
0	1	0	128
0	1	1	256
1	0	0	512
1	0	1	1024
1	1	0	2048
1	1	1	reserved

ADC ClkDiv			MS/s
0	0	0	5,143
0	0	1	4,800
0	1	0	4,235
0	1	1	3,600
1	0	0	2,250
1	0	1	0,973
1	1	0	0,371
1	1	1	0,117

Figure 20: Baseband setup, BB_CONFIG command frame

<u>Format (3 bit) - Data Output Format</u>

Select the data output format. If the Extended Data Mode has been enabled and the RAW bit was enabled in the SYS_CONFIG command frame, the 'Format' field has to be set to the 'mm' option to enable pure raw data output. If the Extended Data Mode has been enabled and the RAW bit was disabled in the SYS_CONFIG command frame, the 'Format' field has to be set to the 'raw A/D' option to enable windowed raw data output.

Table 11: Baseband command frame - Format field

Format (HEX)	Description
0	EXT Data Mode: windowed raw data
1	EXT Data Mode: complex FFT data
2	EXT Data Mode: magnitude/phase of FFT data
3	reserved
4	reserved
5	distance in mm
6 to F	reserved

3.6.1 CFAR Parameters

The constant false alarm rate (CFAR) operator is used to calculate an adaptive threshold above the noise floor. Due to the characteristics of usual target spectra it can be used as an efficient way to achieve a guaranteed detection threshold and reduce false alarms. However, the CFAR operator might not be ideal in every target situation or for every application. It should also be optimized for the specific measurement task.

The used CFAR implementation is the standard cell-averaging (CA-CFAR) approach. The operator is slid through the sample buffer, calculating each cell under test separately. The cell under test is ignored, as well as the number of guard cells left and right of the cell under test. The number of cells left and right of the guard interval are then used to calculate the noise floor around the cell under test. The threshold value is then added to this average.

Due to this approach, the contrast around a clean target is increased. There are, however, situations where the CA-CFAR operator does not achieve sufficient results. This is the case when there are many targets in similar distances. The targets will be treated as noise and may not be detected in such cases. The number of targets in adjacent range cells should therefore not exceed the guard interval value to achieve good results.

CFAR Threshold (5 bit)

CFAR threshold value added to the average of the CA-CFAR operator. The CFAR threshold is a 5 bit unsigned integer value with a value range of 0 to 31.

CFAR Size (4 bit)

The number of cells left and right of the CA-CFAR guard interval are then used to calculate the noise floor around the cell under test. The CFAR size is a 4 bit unsigned integer value with a value range of 0 to 15.

CFAR Grd (2 bit)

The CFAR guard number is the number of guard cells left and right of the cell under test that are ignored for the CA-CFAR calculation. The CFAR guard is a 2 bit unsigned integer value with a value range of 0 to 3.

3.6.2 FFT Parameters

Average N (3 bit)

Average N is a 3 bit unsigned integer value with a value range of 0 to 7. Average N configures the averaging filter at the output of the FFT calculation. The filter is calculated using the following formula, where x is the sample number:

$$output(x) = oldvalue(x) + \frac{(oldValue(x) - newValue(x))}{n_{AVG}}$$

FFT Size (3 bit)

Configures the number of FFT points from 32 to 1024. The FFT size is a 3 bit unsigned integer value. The value range is 0 to 7, interpreted as 2 to the power of (’FFT Size’ + 5). For example, ‘FFT size’ = 0 is interpreted as 2^(0+5) = 32, ‘FFT size’ = 5 is interpreted as 2^(5+5) = 1024.

Downsampling (3 bit)

The downsampling factor is used to decrease the number of samples for the FFT. ‘Downsampling’ is a 3 bit unsigned integer value. The value range is 0 to 7, interpreted as 0 for the value 0 and 2 to the power of (’Downsampling’ - 1) for the values 1 to 7. For example, ‘Downsampling’ = 0 is interpreted as 0, ‘Downsampling’ = 1 is interpreted as 2^(1-1) = 1, ‘Downsampling’ = 7 is interpreted as 2^(7-1) = 64.

A downsampling factor of one will result in half the number of samples, a factor of two in a quarter of the number of samples and so on. To achieve this, each number of ‘Downsampling’ samples are averaged and the remaining sample positions in the sample buffer are zero-padded like in Figure 21.

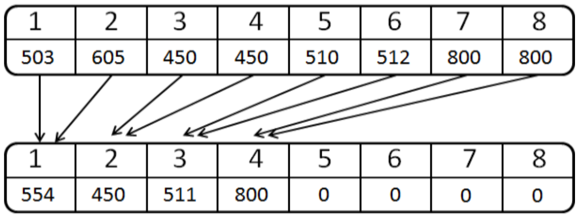

Figure 21: Downsampling and zero padding example for an eight-position sample window

3.6.3 ADC Sampling Parameters

<u>#Ramps - Number of Ramps</u>

‘#Ramps’ determines the number of ramps that are used for every measurement after the pre-measurement phase. ‘#Ramps’ is a 3 bit unsigned integer value. The value range is 0 to 7, interpreted as 2 to the power of ’#Ramps’. For example, ’#Ramps’ = 0 is interpreted as 2^0 = 1, ’#Ramps’ = 7 is interpreted as 2^7 = 128.

Setting this value to 4 will result in 6 ramps chirped for each measurement. 2 ramps are taken as pre-measurement for the automatic gain detection and 4 ramps will be A/D converted and sample-wise added before the FFT is calculated.

<u>#Samples - Number of Samples</u>

‘#Samples’ is the number of samples taken per ramp. The number of samples is a 3 bit unsigned integer value. The value range is 0 to 7, interpreted as 2 to the power of (‘#Samples’ + 5). For example, ‘#Samples’ = 0 is interpreted as 2^(0+5) = 32, ‘#Samples’ = 5 is interpreted as 2^(5+5) = 1024.

Increase this number to achieve better resolution and to get longer ramp times. The ramp time is rounded to microsecond resolution. The ramp time is calculated using the following formula:

$$T_{ramp} = \frac{N_{samples}}{F_{ADC}}$$

<u>ADC ClkDiv - ADC clock divider / Sample frequency</u>

‘ADC ClkDiv’ determines the ADC clock divider setting. ‘ADC ClkDiv’ is a 3 bit unsigned integer value. The value range is 0 to 7, according to the index of an internal look-up table which leads to the given number of MS/s according to the ‘ADC ClkDiv’ table in Figure 20.

Higher values result in longer A/D conversion times and thus longer ramp times and can increase the signal strength of low signals.

3.6.4 Default Settings

Below you can find the default settings of the WebGUI for the PLL_CONFIG command frame to speed up your development by simple copy and paste into your favorite terminal program.

WebGUI Default Setting (SiRad Easy® with 24 GHz frontend):

!BB034C125	

WebGUI Default Setting (SiRad Simple® and Easy® with 122 GHz frontend):

!BB034C125	

3.7 Special Function Commands

Certain commands, explained in this section, use only a single letter to execute a function very fast. Send these commands three times in a row in case they are not executed.

Get fill error info - !E

Request a detailed error info frame at the next transmission slot.

Get system info - !I

Request a system info frame at the next transmission slot.

Do frequency scan - !J

The system scans the maximum usable bandwidth of the installed frontend at every startup. To trigger that scan manually at runtime, use this command.

Set to max. bandwidth - !K

Set the ramp bandwidth to the previously measured maximum.

Send Pre-Trigger - !L

Software command to trigger the pre-measurement. The pre-measurement consists of 2 ramps which are used to evaluate the maximum usable baseband gain by the Auto Gain Control (AGC) Mode feature. It is executed before each measurement. If the Pre-trigger bit in the SYS_CONFIG register is set and the Self-trigger bit is reset, the system waits for an external Pre-trigger and trigger either on the hardware trigger line or this software command.

Send Trigger - !M

Software command to trigger a measurement. Execute this command max. 40 ms after the Pre-trigger, or the system will go back to idle.

Send Pre-Trigger+Trigger - !N

Software command to send a Pre-trigger (!L) and a trigger (!M) all in one command.

Get version info - !V

Request a version info frame at the next transmission slot.

4 Extended Data

Figure 22 shows the supported extended data frames and Table 12 lists their purpose. The extended data frames can be configured to transmit the raw data of the AD converter. The blue parts in Figure 22 indicate start and stop markers, frame identifiers and delimiters as well as signs, orange and green parts indicate data parts.

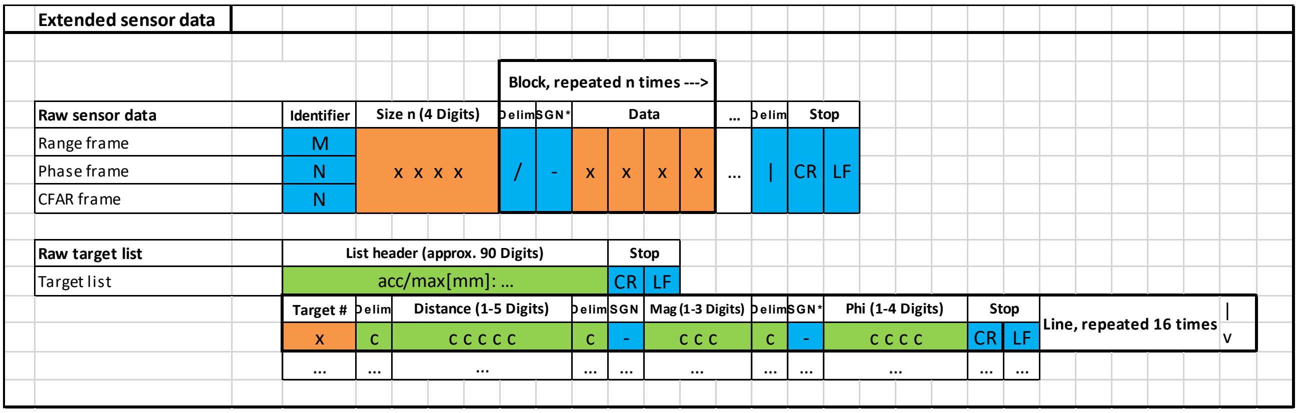

Figure 22: Extended data frames overview

Table 12: Extended data frames overview, description

Extended data frame	Description
Range frame	Contains distance data extracted from the FFT output
Phase frame	Contains phase information extracted from the FFT output
CFAR frame	Contains the output of the CFAR operator
Target list frame	Contains the target list with the detected targets

The extended data output may contain M-frames that transmit the range data, N-frames that transmit the phase and CFAR data, as shown in the example in Figure 23. The standard frames for the status update !U, version info !V, system info !I, and the error frames !E, can be used together with the extended data output mode.

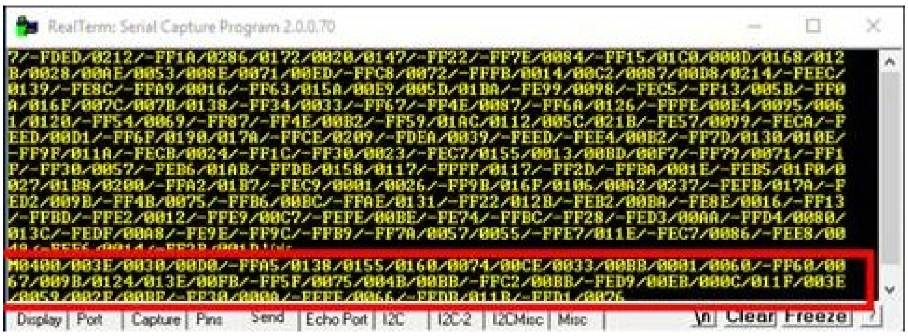

Figure 23: Extended data frame (M frame marked) in a terminal window

To enable the extended data or raw data output, the following settings should be made using the commands described in Section 3. The procedure is explained in detail in the steps below.

1. Disable downsampling

Set the ‘Downsampling’ Bits (10, 11, 12) in the BB_CONFIG command to ‘000’ under consideration of your other desired settings (marked with an ‘x’) like in Figure 24. The default settings for quick copy and paste into your favorite terminal program are shown in Figure 25.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x

Figure 24: BB_CONFIG command mask to disable the downsampling

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
DEFAULT	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1
HEX	B				0				3				4				C				1				2				5			

Figure 25: BB_CONFIG default & downsampling off: !BB034C125

2. Activate the Extended Data Mode

Set the EXT Bit (10) in the SYS_CONFIG command to ‘1’ and the ST Bit (9), TL Bit (8), P Bit (7), C Bit (6) and the R Bit (5) to ‘0’ under consideration of your other desired settings (marked with an ‘x’) like shown in Figure 26. The default settings for quick copy and paste into your favorite terminal program are shown in Figure 27. Proceed with either 3a or 3b to select the data output.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	x	x	x	x

Figure 26: SYS_CONFIG command mask to enable the Extended Data Mode

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
DEFAULT EASY	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0
HEX	0				1				0				0				4				A				0				A			
DEFAULT SIMPLE	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	1	0
HEX	0				1				0				0				4				6				0				A			

Figure 27: SYS_CONFIG default & EXT mode on: !S01004A0A (Easy), !S0100460A (Simple)

3a. Select the kind of FFT data output

To enable the FFT data output, set the RAW Bit (17) in the SYS_CONFIG frame to ‘0’ and enable the FFT distance data, phase data and/or the target list output by setting the R Bit (5) for the distance data, the C Bit (6) for the CFAR data, the P Bit (7) for the phase data, the TL Bit (8) for the target list data, and/or the ST Bit (9) for the status update data in the SYS_CONFIG frame to ‘1’ under consideration of your other desired settings (marked with an ‘x’) like shown in Figure 28. The default settings for quick copy and paste into a terminal program are shown in Figure 29.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	1	X	X	X	X	X	x	x	x	x

Figure 28: SYS_CONFIG command mask to enable FFT data output

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
DEFAULT EASY	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	1	0
HEX	0				1				0				0				4				A				5				A			
DEFAULT SIMPLE	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0	1	0
HEX	0				1				0				0				4				6				5				A			

Figure 29: SYS_CONFIG default & FFT data output: !S01004A5A (Easy), !S0100465A (Simple)

Further, to enable the complex or magnitude/phase FFT data output, select the according Format Bits (30, 31, 32) in the BB_CONFIG frame under consideration of your other desired settings (marked with a lower case 'x') like shown in Figure 30. For complex FFT data, set the Format Bits to 'FFT comp' (001). The default settings for quick copy and paste into your favorite terminal program are shown in Figure 31. For magnitude/phase FFT data, set the Format Bits to 'FFT mag/ph' (010). The default settings for quick copy and paste into your favorite terminal program are shown in Figure 32.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
	X	X	X	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x

Figure 30: BB_CONFIG command mask to select FFT data output

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
DEFAULT	0	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1
HEX	3				0				3				4				C				1				2				5			

Figure 31: BB_CONFIG default & complex FFT data output: !B3034C125

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
DEFAULT	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1
HEX	5				0				3				4				C				1				2				5			

Figure 32: BB_CONFIG default & magnitude/phase FFT data output: !B5034C125

3b. Select the kind of RAW data output

To enable the windowed or unwindowed raw data output, set the RAW Bit (17) in the SYS_CONFIG frame to '1' under consideration of your other desired settings (marked with an 'x') like shown in Figure 33. The default settings for quick copy and paste into your favorite terminal program are shown in Figure 34.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x	1	0	0	0	0	0	x	x	x	x

Figure 33: SYS_CONFIG command mask to enable raw data output

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	resl	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
DEFAULT EASY	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0
HEX	0				1				0				1				4				A				0				A			
DEFAULT SIMPLE	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	1	0	1	0
HEX	0				1				0				1				4				6				0				A			

Figure 34: SYS_CONFIG default & raw data output: !S01014A0A (Easy), !S0101460A (Simple)

For unwindowed raw data, set the Format Bits (30, 31, 32) in the BB_CONFIG frame to 'dist mm' (101) under consideration of your other desired settings (marked with an 'x') like in Figure 35. The default settings for quick copy and paste a terminal program are shown in Figure 36.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
	1	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x

Figure 35: BB_CONFIG command mask to enable unwindowed raw data output

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
DEFAULT	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1
HEX	B				0				3				4				C				1				2				5			

Figure 36: BB_CONFIG default & unwindowed raw data output: !BB034C125

The size of the unwindowed raw data output is always 2 * '#Samples'.

For windowed raw data, set the Format Bits (30, 31, 32) in the BB_CONFIG frame to 'raw A/D' (000) under consideration of your other desired settings (marked with an 'x') like in Figure 37. The default settings for quick copy and paste into a terminal program are shown in Figure 38.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x

Figure 37: BB_CONFIG command mask to enable windowed raw data output

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	Format			CFAR Threshold [dB]					CFAR Size				CFAR Grd		Average n			FFT Size			Downsampling			# Ramps			# Samples			ADC ClkDiv		
DEFAULT	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1
HEX	1				0				3				4				C				1				2				5			

Figure 38: BB_CONFIG default & windowed raw data output: !B1034C125

The size of the windowed raw data output is always 2 * 'FFT Size'.

<u>5. Adjust the gain settings (optional)</u>

Preferably disable the Auto Gain Control (AGC) Mode and set a suitable gain manually by setting the AGC Bit (15) in the SYS_CONFIG command to '0' and choose a setting from below for the 'Gain' Bits (13, 14) like shown in Figure 39, but this is not mandatory. With the AGC Mode disabled, no additional ramps are measured before the actual distance measurement, which speeds up the frame rate.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	res	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	0	X	X	x	x	1	0	0	0	0	0	x	x	x	x

Figure 39: SYS_CONFIG command mask to disable Auto Gain Control Mode

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							RAW	res	AGC	Gain		SER2	SER1	EXT	ST	TL	P	C	R	DC	res	SLF	PRE
DEFAULT EASY	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0
HEX	0				1				0				1				3				A				0				A			
DEFAULT SIMPLE	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0	0	0	0	0	1	0	1	0
HEX	0				1				0				1				3				6				0				A			

Figure 40: SYS_CONFIG default & manual gain: !S01013A0A (Easy), !S0101360A (Simple)

5 Disclaimer / License

By using the information in this document, our products and the provided software, you agree to our disclaimer and license conditions.

| Disclaimer |

==========

The information contained in this document is subject to change at any time without notice.

Silicon Radar GmbH assumes no responsibility or liability for any loss, damage or defect of a product that is caused in whole or in part by

(i) use of any circuitry other than circuitry embodied in a Silicon Radar GmbH product,

(ii) misuse or abuse including static discharge, neglect or accident,

(iii) unauthorized modification or repairs which have been soldered or altered during assembly and are not capable of being tested by Silicon Radar GmbH under its normal test conditions, or

(iv) improper installation, storage, handling, warehousing or transportation, or

(v) being subjected to unusual physical, thermal, or electrical stress.

Silicon Radar GmbH makes no warranty of any kind, express or implied, with regard to this material, and specifically disclaims any and all express or implied warranties, either in fact or by operation of law, statutory or otherwise, including the implied warranties of merchantability and fitness for use or a particular purpose, and any implied warranty arising from course of dealing or usage of trade, as well as any common-law duties relating to accuracy or lack of negligence, with respect to this material, any Silicon Radar product and any product documentation. Products sold by Silicon Radar are not suitable or intended to be used in a life support applications or components to operate nuclear facilities, or in other mission critical applications where human life may be involved or at stake. All sales are made conditioned upon compliance with the critical uses policy set forth below.

CRITICAL USE EXCLUSION POLICY: BUYER AGREES NOT TO USE SILICON RADAR GMBH'S PRODUCTS FOR ANY APPLICATION OR IN ANY COMPONENTS USED IN LIFE SUPPORT DEVICES OR TO OPERATE NUCLEAR FACILITIES OR FOR USE IN OTHER MISSION-CRITICAL APPLICATIONS OR COMPONENTS WHERE HUMAN LIFE OR PROPERTY MAY BE AT STAKE.

Silicon Radar GmbH owns all rights, title and interest to the intellectual property related to Silicon Radar GmbH's products, including any software, firmware, copyright, patent, or trademark. The sale of Silicon Radar GmbH products does not convey or imply any license under patent or other rights. Silicon Radar GmbH retains the copyright and trademark rights in all documents, catalogs and plans supplied pursuant to or ancillary to the sale of products or services by Silicon Radar GmbH. Unless otherwise agreed to in writing by Silicon Radar GmbH, any reproduction, modification, translation, compilation, or representation of this material shall be strictly prohibited.

Silicon Radar GmbH, 2019

| License |

========

The radar software delivered with the Install Package is provided by Silicon Radar GmbH.

The software is provided ‘as is’ for testing purposes only. While hoping it will be useful, there is NO WARRANTY of any kind, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no event shall the distributor, the authors, any of the contributors, or any third party be liable for any direct, indirect, incidental, or consequential damages (including, but not limited to, procurement of substitute goods and services, loss of use, damage to hardware, loss of data, loss of profits, or business interruption).

All software in the Install Package marked as third-party is copyrighted by their respective owners and covered by their respective licenses. Please see the corresponding websites or the package information for details. Please also see the README.txt files of the Install Package for details.

Silicon Radar GmbH, 2019