



Radar System Laboratory

A.A. 2022/2023

*Master Degree in Telecommunication Engineering
University of Pisa*

 francesco.mancuso@phd.unipi.it



Activities Overview

FMCW Radar Lab Measurements

- EVALKIT SiRad Simple®
- Board configuration
- MATLAB acquisition
- A little bit of processing





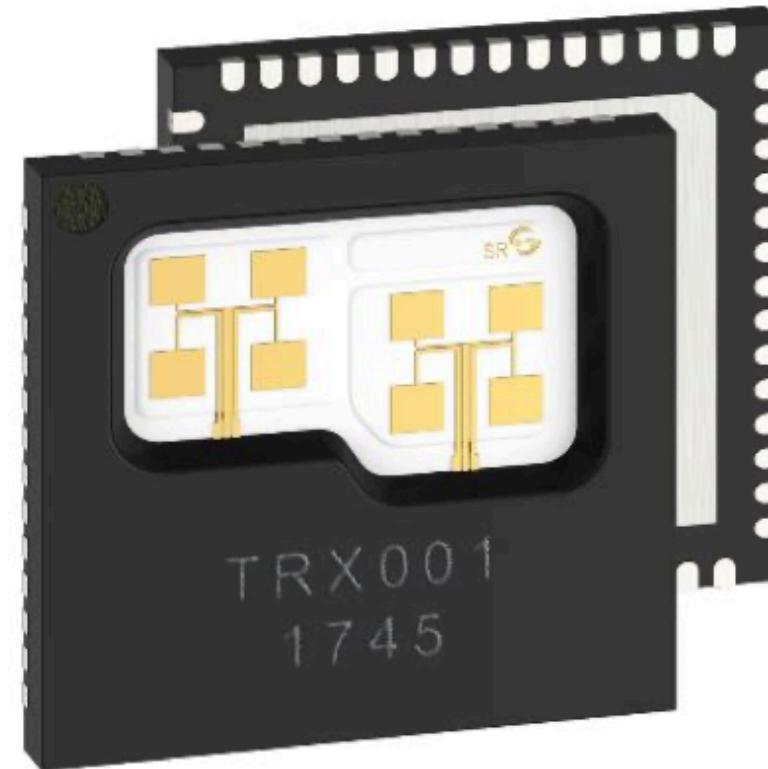
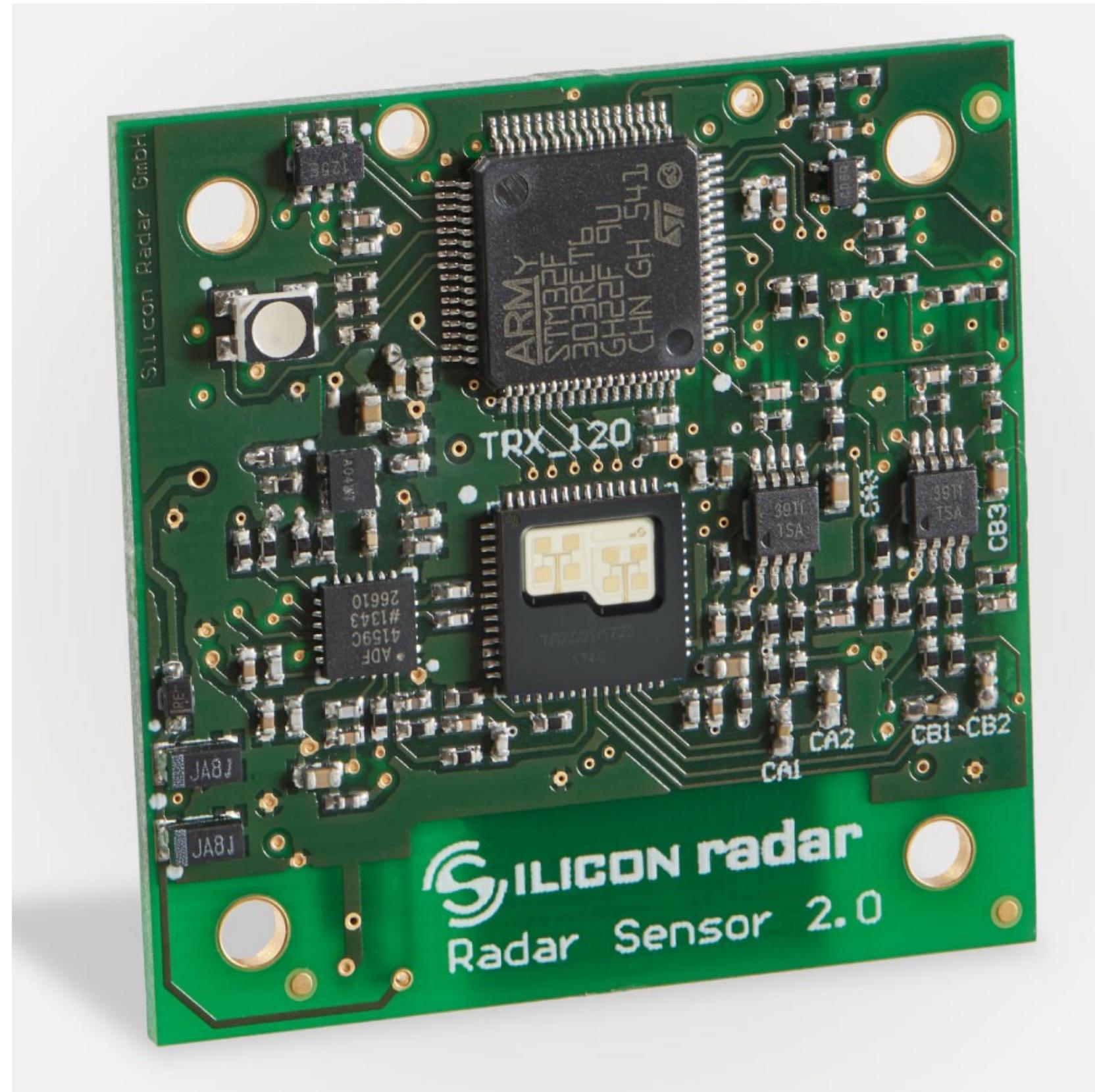
EVALKIT SiRad Simple®

120GHz radar evaluation kit

...recently deprecated 😞



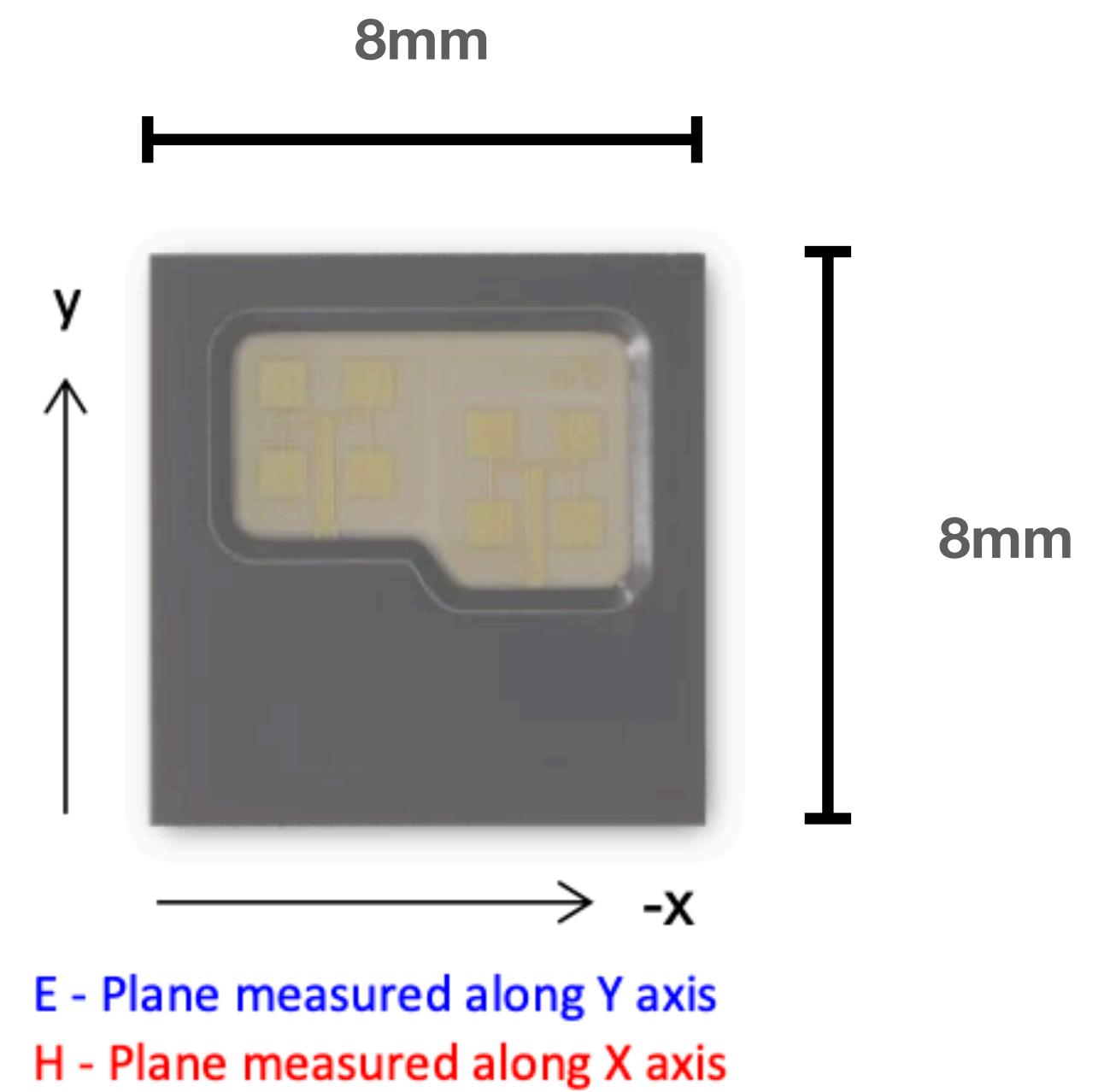
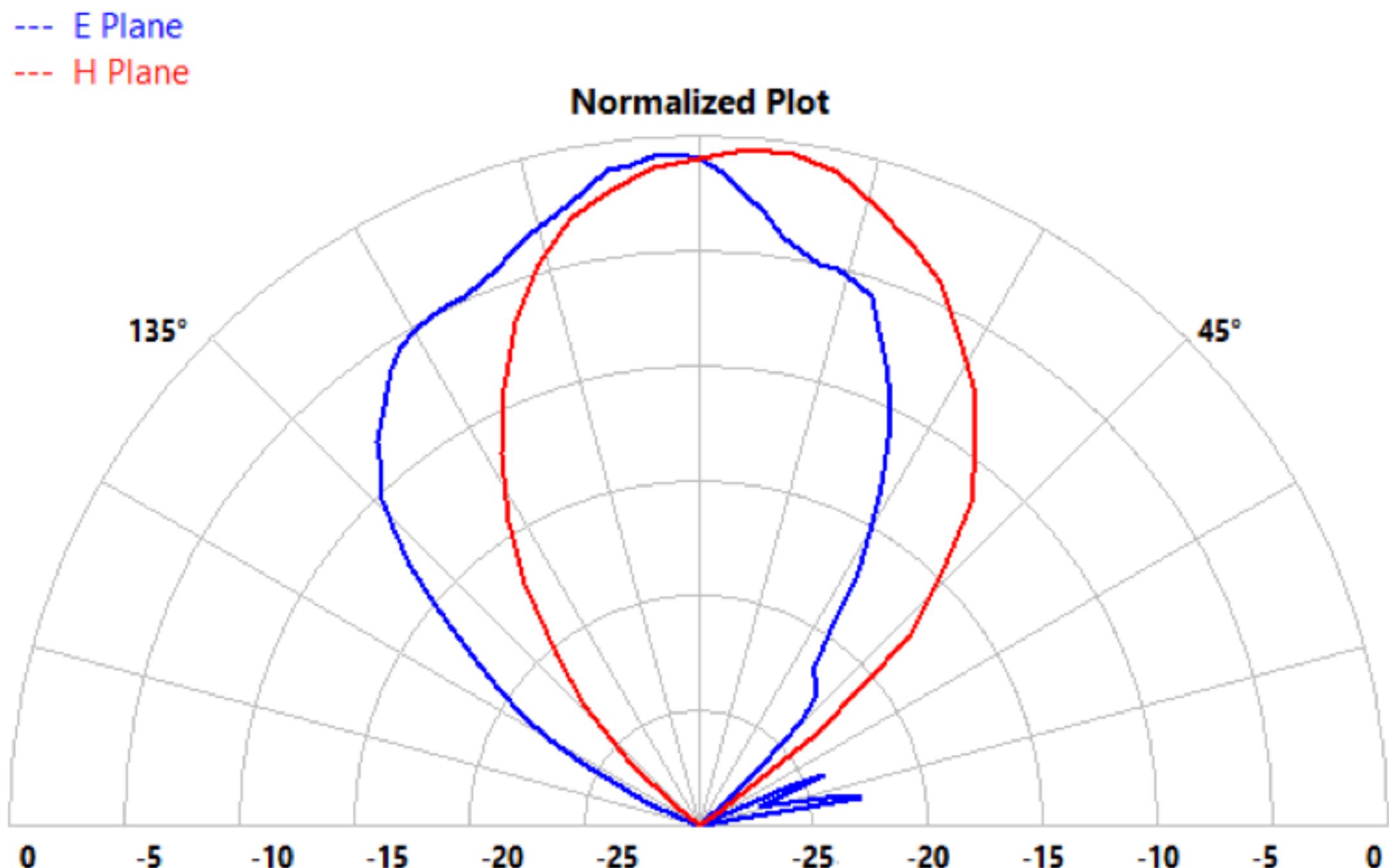
TRX_120_001 (1)



- Radar front end (RFE) with antennas in package for 122-GHz ISM band
- Wide bandwidth of up to 6 GHz
- Single supply voltage of 3.3 V

https://siliconradar.com/datasheets/Datasheet TRX_120_001_V1.4.pdf

TRX_120_001 (2)

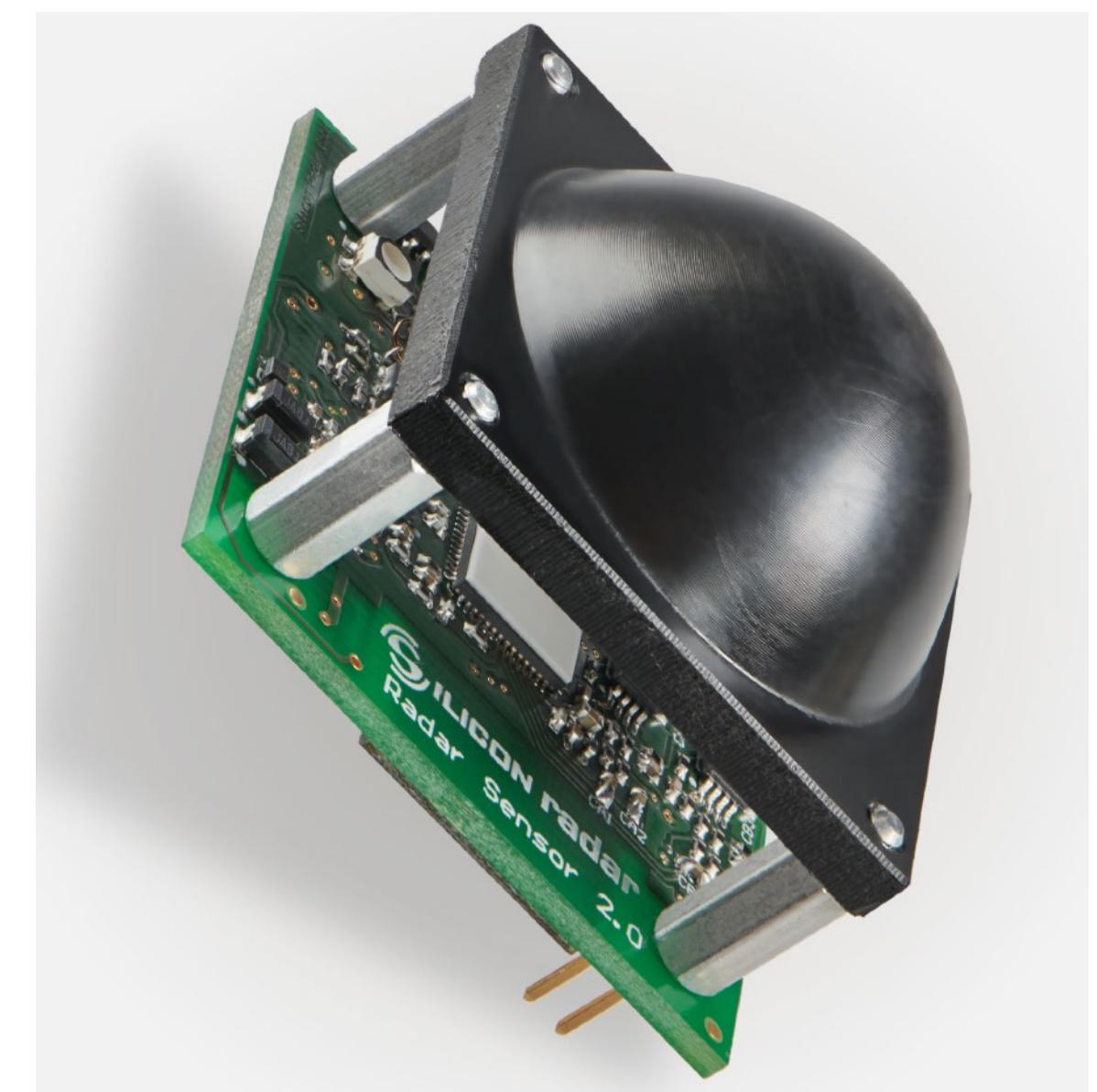
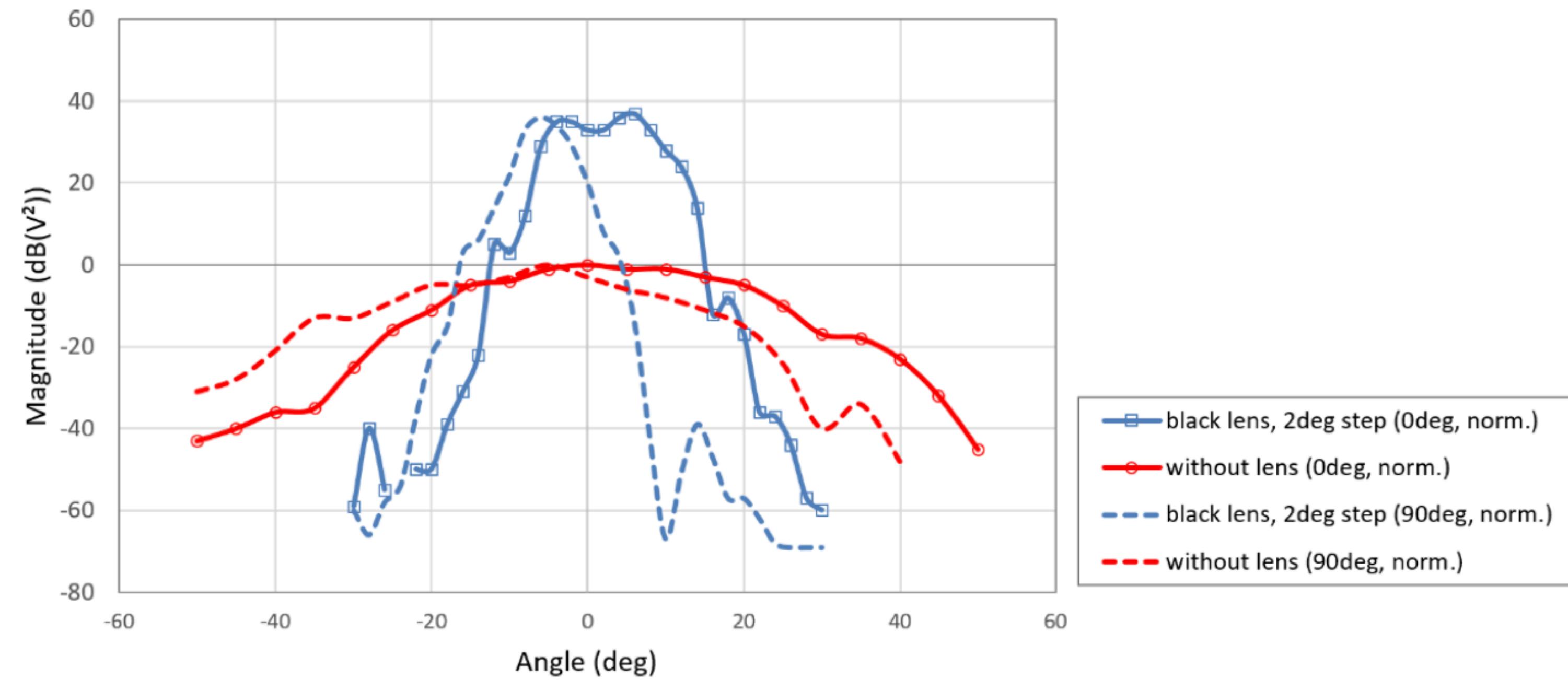


Combined Radiation Pattern Measurements of TX and RX Patch Antennas

https://siliconradar.com/datasheets/Datasheet_TRX_120_001_V1.4.pdf

Collimator lens

- Optional accessory for evaluation kit **SiRad Simple®**, it focuses the beam of the radar front end



$\pm 4^\circ$ @ -6dB

<https://siliconradar.com/datasheets/Datasheet Collimator Lens V1.1.pdf>

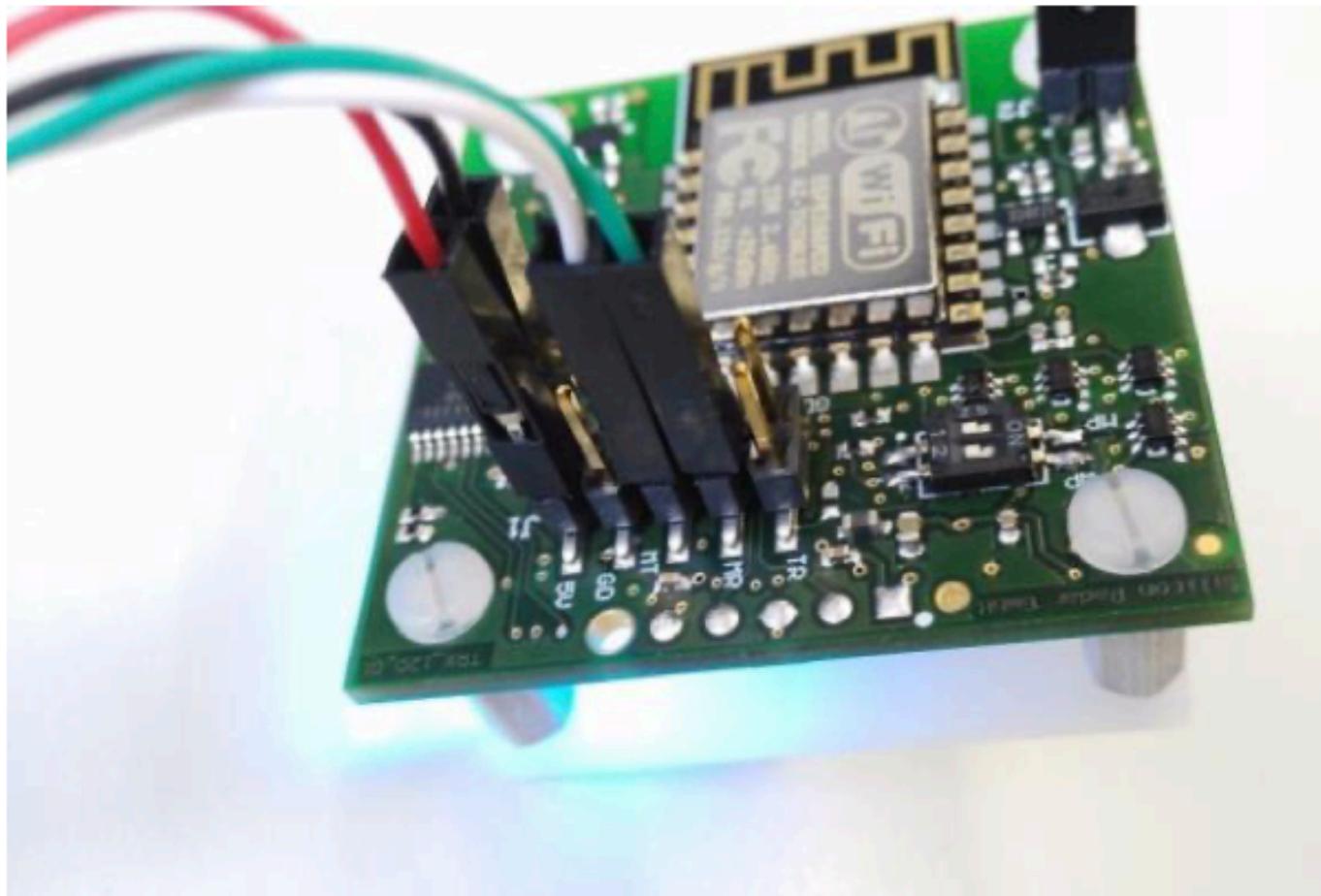
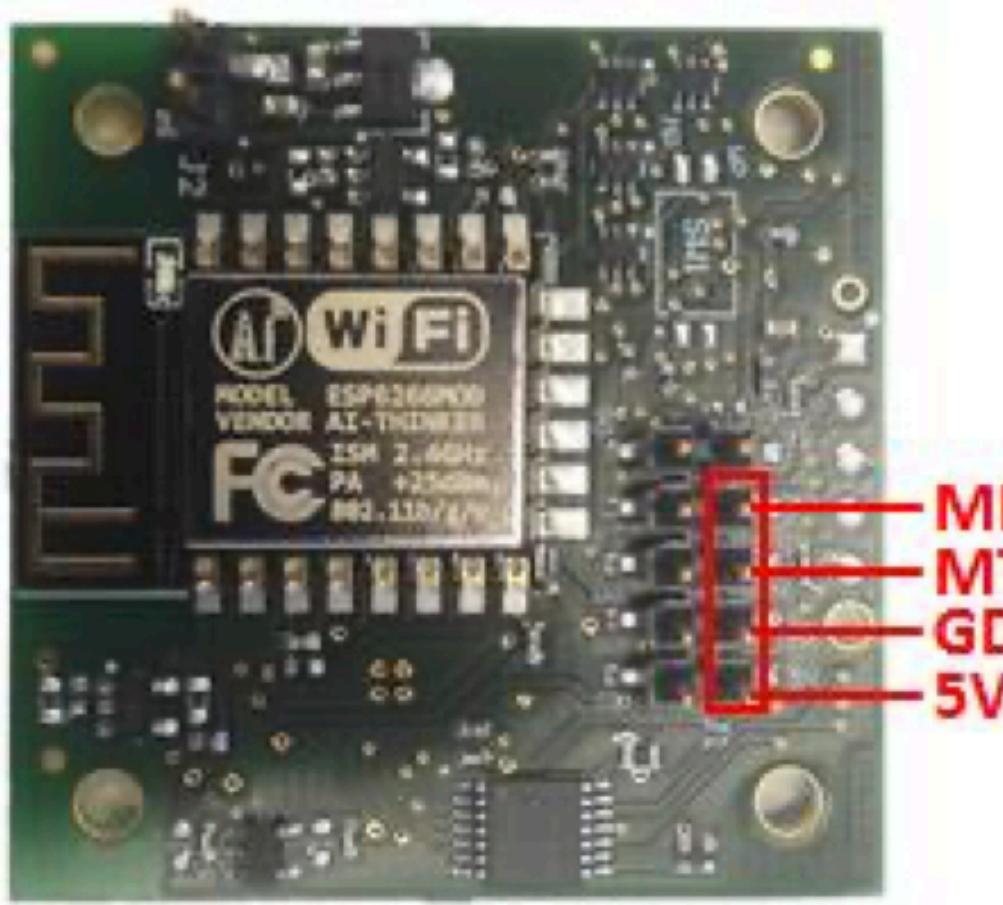
01011
11010
01011



Board configuration

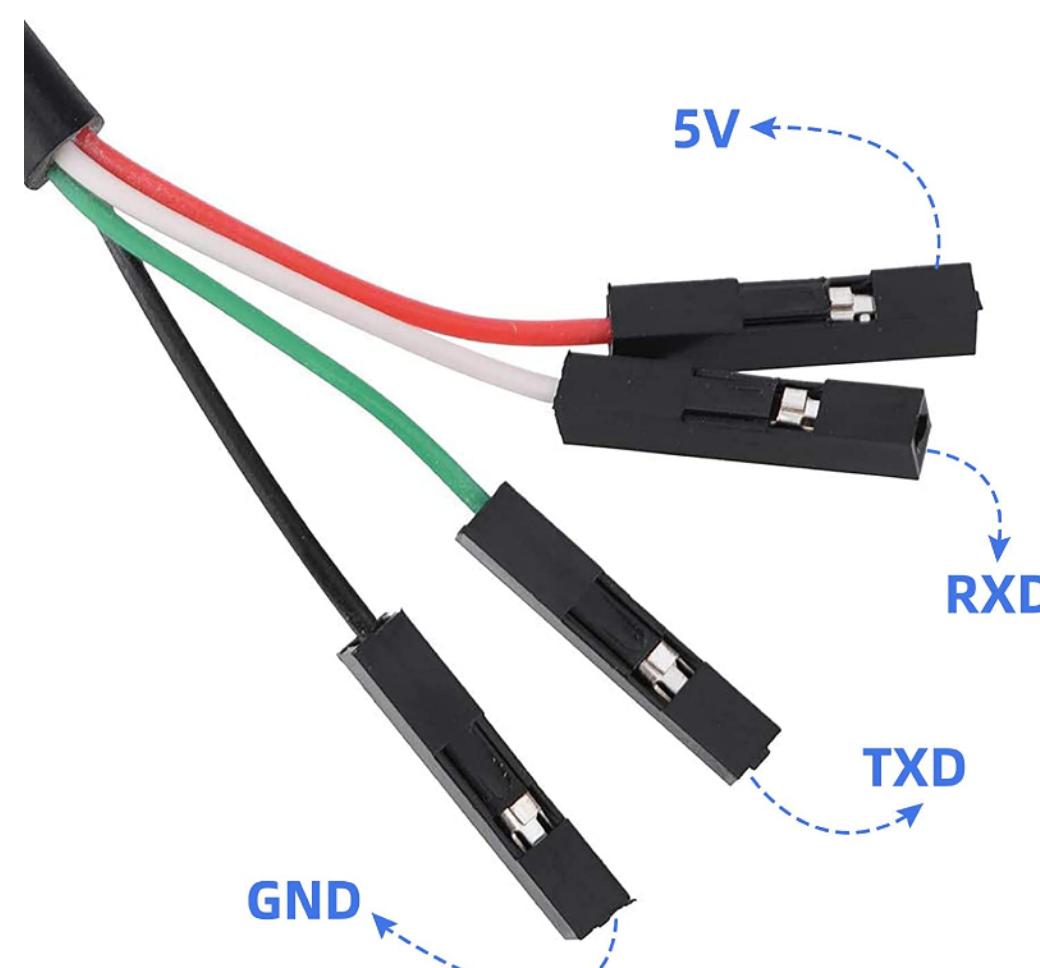
USB-to-UART Bridge

Serial/USB data connection



FTDI cable Board

5V	→	5V
GND	→	GD
TXD	→	MR
RXD	→	MT



Make sure that both DIP switches are in their **OFF** positions and the power jumper (J2) for the WiFi module is **open** (switched off).

System configuration

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	res	LOG	FMT	LED	reserved			Protocol	AGC	Gain	SER2	SER1	ERR	ST	TL	C	R	P	CPL	RAW	res	res	SLF	PRE							
SelfTrigDelay			LED						SER2			data frames			off		on		SLF													
0 0 0			0 ms			0 0 off			0 output on SER2 off			RAW raw ADC			0		1		0 external trig mode													
0 0 1			2 ms			0 1 1st target			1 output on SER2 on			CPL cmplx FFT			0		1		1 standard mode													
0 1 0			4 ms			1 0 reserved			P phase			R magnitude			0		1															
0 1 1			8 ms			1 1 reserved			C CFAR			TL target list			0		1		0 standard mode													
1 0 0			16 ms						SER1			ST status			0		1		1 use pre-trigger													
1 0 1			32 ms			Gain			0 output on SER1 off			ERR error			0		1		PRE													
1 1 0			64 ms			FMT			0 TL mm distance			AGC			0		1		0 auto gain control off													
1 1 1			128 ms			LOG			1 TL cm distance						0 0 0		WebGUI															
						0 log MAG			0 0 1			0 1 0			TSV output		BIN output		X X X			reserved										

*Reserved bit will have 0 as default value

000 X 1 0 00 XXXX 001 0 00 0 1 00000001 X X 1 0

!S 0000 1000 0000 0010 0001 0000 0001 0010 \r\n



'!S08021012\r\n'

<https://siliconradar.com/datasheets/Protocol Description Easy Simple V2.4.pdf>

PLL configuration

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	1	1	1	1	1	1	1	1	1	1	1	1	1	-2 MHz		
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	-4 MHz		
	...																															
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-65536 MHz		
	0	0	0	0	0	0	0	0	0	0	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	0	0	0	0	0	0	0	1	1	1	1	2 MHz		
	...																															
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+65534 MHz		

Format Field	Field Size	Description
Bandwidth	16 bits	Negative values result in a falling ramp slope, positive values in a rising saw tooth shape; representation is in two's complement

XXXXXXXXXXXXXX 0000000111110100

BW = 1 GHz

!P 0000 0000 0000 0000 0001 1111 0100 \r\n



'!P000001F4\r\n'

<https://siliconradar.com/datasheets/Protocol Description Easy Simple V2.4.pdf>

*Reserved bit will have 0 as default value

Baseband configuration

	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	WIN	FIR	DC	CFAR	CFAR Threshold [dB]				CFAR Size				CFAR Grd		Average n		FFT Size			Down sample			# Ramps		# Samples			ADC ClkDiv					
WIN					FFT Size				Down sample				# Ramps		# Samples			ADC ClkDiv															
0	windowing off				0	0	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	2,571				
1	windowing on				0	0	1	64	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	64	0	0	1	2,400				
FIR					0	1	0	128	0	1	0	2	0	1	0	4	0	1	1	8	0	1	0	1	128	0	1	0	2,118				
0	FIR filter off				0	1	1	256	0	1	1	4	0	1	1	8	0	1	1	16	0	1	1	1	256	0	1	1	1,800				
1	FIR filter on				1	0	0	512	1	0	0	8	1	0	0	16	1	0	0	32	1	0	0	0	512	1	0	0	1,125				
DC					1	0	1	1024	1	0	1	16	1	0	1	32	1	0	1	64	1	0	1	1	1024	1	0	1	0,487				
0	DC cancellation off				1	1	0	2048	1	1	0	32	1	1	1	64	1	1	1	128	1	1	1	1	2048	1	1	0	0,186				
1	DC cancellation on				1	1	1	reserved	1	1	1	64	1	1	1	128	1	1	1	1	1	1	1	1	reserved	1	1	1	0,059				
CFAR					CFAR Threshold [dB]				dB				CFAR Size			CFAR Grd			Average n														
0	0	CA-CFAR				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0	1	CFAR_GO				0	0	0	1	2	0	0	0	1	1	0	1	0	1	0	1	0	1	1	0	1	1	0	1	1			
1	0	CFAR_SO							
1	1	reserved				1	1	1	1	30	1	1	1	1	1	15	1	1	1	1	1	1	1	1	3	1	1	1	3	1	1		

<https://siliconradar.com/datasheets/Protocol Description Easy Simple V2.4.pdf>

MATLAB acquisition

```
s = serialport("COM3", 1000000);
```



Serial port configuration

```
sPort = serialportlist; .....  
baudRate = 1e6;  
  
serialPort = sPort(2); .....  
fprintf("Connecting to %s ...\\n", serialPort)  
com_port = serialport(serialPort, baudRate);  
  
set(com_port, "DataBits", 8);  
set(com_port, "Parity", 'none');  
set(com_port, "StopBits", 1);  
set(com_port, "Timeout", 1);  
configureTerminator(com_port, 'CR/LF');
```

serialportlist() lists all the serial devices

On my PC, it was the second element;
this must be adapted

Configuration of the serial port

Note for Linux users: I had some problems with Ubuntu, the OS wanted all the permissions to the serial ports. Type in Terminal: `sudo chmod 666 /dev/tty*`

<https://www.mathworks.com/help/matlab/ref/serialport.html>

Data acquisition

External trigger

```
while true
    writeline(com_port, "!N");
    buf = '';
    while isempty(buf) || ~startsWith(buf, '!M')
        buf = readline(com_port);
    end

    frameM = split(buf);
    frameM = frameM(4:end);

    dataM = str2double(frameM);
    dataI = dataM(1:2:end-1);
    dataQ = dataM(2:2:end);

    complexData = (dataI + 1i*dataQ).';
    /--- GENERIC PROCESSING ---/
end
```

Trigger acquisition

Since it may happen that the received buffer is empty, or it does not start with the right control character (i.e., error frame, information frame), the `readline()` is repeated until the data is acquired properly

The I samples are on odd indices, while the Q samples are on even indices



A little bit of processing

Range profiles evaluation

GitHub repo Radar-Systems-Lab

The screenshot shows the GitHub repository page for 'M-M-Lab/Radar-Systems-Lab'. The repository is public and contains 65 commits from the 'main' branch. The commits are listed in descending order of age, starting with 'Delete firstScript.m' by 'mandugo' 2 hours ago. Other commits include 'update readme with CFAR detector information' 3 months ago, 'Added DC correction to default config' 4 months ago, and 'Initial commit' 5 months ago. The repository also includes files like LICENSE.md, README.md, and SiliconRadar.pdf. The 'About' section describes it as a Python GUI for EVALKIT SiRad Simple for real time applications and data recording, and MATLAB code for offline processing. It features labels for signal-processing, radar, and fmcw-radar. The 'Releases' section indicates no releases have been published, and the 'Packages' section is empty.

<https://github.com/M-M-Lab/Radar-Systems-Lab>



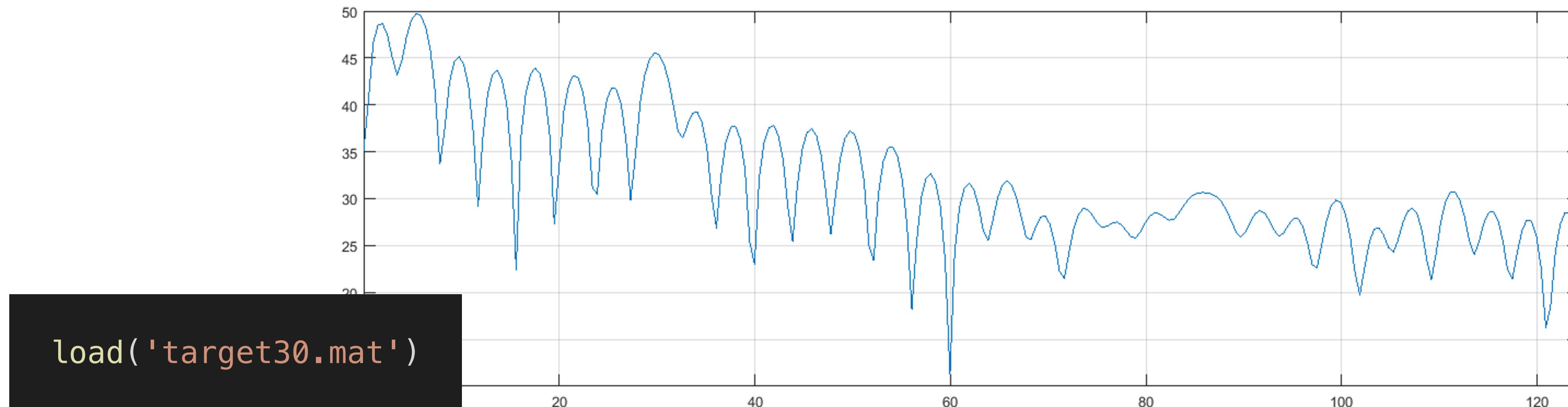
<https://github.com/M-M-Lab/Radar-Systems-Lab>

Radar-Systems-Lab/firstLesson

- First example: range profile evaluation
- Second example: live range profile plot
- Third example: basic signal processing

.../firstLesson/firstExample mlx

```
if length(complexData) == nSamples
    rangeProfile = fft(complexData,nfft);
    curveToPlot = abs(rangeProfile(1:end/2));
    [m,i] = max(curveToPlot);
    set(h(1),'YData',20*log10(curveToPlot));
    set(h(2),'XData',rangeVec(i),'YData',mag2db(m),'Marker','o','Color','r');
    title('Range Profile',strcat("maximum at ",num2str(round(rangeVec(i),2))," cm"))
end
```



.../firstLesson/secondExample.mlx

```
if length(complexData) == nSamples
    rangeProfile = fft(complexData,nfft);
    curveToPlot = abs(rangeProfile(1:end/2));
    [m,i] = max(curveToPlot);
    set(h(1),'YData',20*log10(curveToPlot));
    set(h(2),'XData',rangeVec(i),'YData',mag2db(m),'Marker','o','Color','r');
    title('Range Profile',strcat("maximum at ",num2str(round(rangeVec(i),2))," cm"))
end
```

Same processing as firstExample.mlx, but live: the range profile is updated for every received frame and the board uses its internal trigger.

.../firstLesson/thirdExample.mlx

```
windowHann = hann(nSamples).';  
rangeProfileW Hann = fft(complexData.*windowHann,nfft);  
curveToPlotW Hann = abs(rangeProfileW Hann(1:end/2));  
  
figure  
plot(rangeVec,20*log10(curveToPlot))  
hold on  
plot(rangeVec,20*log10(curveToPlotW Hann))
```

