Action TU1208
Civil Engineering Applications
of Ground Penetrating Radar

Final Conference

Warsaw, Poland
25-27 September 2017

National Institute
of Telecommunications
of Poland

Guide to building a GPR radar for educational use

Sections: RF Hardware design, Firmware of microcontroller, and management software.

Vincenzo Ferrara (Italy), Margarita Chizh (Russia), Andrea Pietrelli (Italy)
(email : vincenzo.ferrara@uniroma1.it
mchizh@rslab.ru
andrea.Pietrelli@uniroma1.it)

COST is supported by the
EU Framework Programme Horizon2020

Talk Layout

- **Preamble**
- **Basic principles**
- **Guide**
  - RF Hardware design
  - Software and Firmware to manage microcontroller
  - Basic software for interpreting output signal
- **Recommendations**
- **Conclusions**
- **References**
- **Additional Material**
  - List of PCB layout files
  - List of datasheets
  - Software and firmware codes list
- **Biographies** and **contact details** of **Authors**

COST is supported by the
EU Framework Programme Horizon2020

Preamble

Aims and project teams

Preamble

Among the objectives of COST Action TU1208, the dissemination of the technologies used to build the ground-penetrating radar (GPR) should be a start point for introducing young people into the understanding and use of this detection instrument.

The inspiration for this project was the radar course arranged by Lincoln Laboratory at Massachusetts Institute of Technology (MIT) [6], increasing performance and interactivity of that device.

The purpose and hope is that creating a cheap GPR and introducing it into the educational process will promote a more wide use of this effective non-invasive and non-destructive technique.

This document aims at explaining the design of a frequency-modulated continuous-wave (FCMW) GPR, describing step by step construction phases and providing schemes, software codes, and all the documentation needed to build a GPR.

So, the present guide aims at enhancing the overall technical literacy of students, enabling them to develop their own radar using minimum prerequisite knowledge of the radar.

Preamble

The GPR radar project is made up of several parts. Shortly[3][5]:

- RF electronic hardware design.
- Antennas design.
- DC power source design.
- Software development for implementing firmware into microcontrollers.
- Software development for managing device by PC.
- Software development for interpreting the GPR signal detection.

Preamble

This document is a guide to carry out:

- RF electronic hardware design.
- DC power source design.
- Software development for implementing firmware into microcontrollers.
- Software development for managing device by PC.
- Software development for interpreting the GPR signal detection.

Designing appropriate antennas, suited for use in the GPRs, will be the subject of another document.

Preamble

Teams that participated in the project phases

project phase	Laboratory
• RF electronic hardware design. • DC power source design.	Electronics for the Environment Laboratory – Dep. of Information Eng., Electronics and Telecommunications (DIET) – Sapienza University of Rome
• Software development for implementing firmware into microcontrollers. • Software development for managing device by PC.	Remote Sensing Laboratory Bauman Moscow State Technical University & Electronics for the Environment Laboratory - DIET – Sapienza University of Rome
• Software development for interpreting the GPR signal detection.	Electronics for the Environment Laboratory - DIET – Sapienza University of Rome

COST is supported by the
EU Framework Programme Horizon2020

Preamble

This guide aims to describe the steps for building a GPR with the main following features:

- FMCW radar type
- Use of modular subsystems that allow you to replace components for implementing design alternatives, and get greater understanding in the case of educational activity
- Device managed by PC for interactive control of GPR
- Frequency sweep: f_{min}=1.3 GHz, f_{max}=2.6 GHz
- Choice of smaller interval for frequencies of use
- Choice of modulation wave shape
- Choice of modulating wave period
- COTS (components off-the-shelf)
- Low cost

COST is supported by the
EU Framework Programme Horizon2020

Basic principles

Basic principles

Introduction

FMCW radar working is based on the continuous emission of sinusoidal waveform (CW) whose frequency is suitably modulated (FM).

Like CW radar, FMCW radar uses Doppler effect, and can measure the speed of detected mobile objects. Moreover, FMCW radar exceeds the limitation of CW radar, regarding the possibility to measure distance of detected stationary objects, due to missing time reference.

FMCW radar transmits signal which change periodically in the frequency. When radar receives a signal of different frequency from transmitted signal at that instant, it gets a delay Δt correlated to the runtime shift.

COST is supported by the
EU Framework Programme Horizon2020

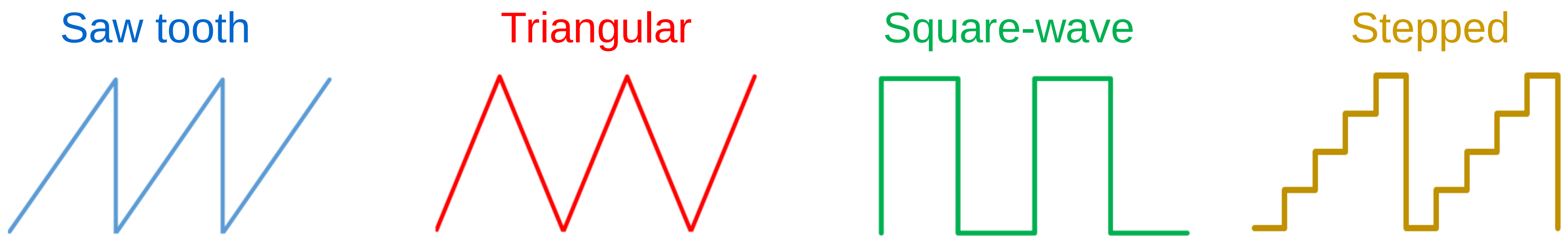
Basic principles

Essential characteristics

So, basic design FMCW radar shows:

- The adoption of local oscillator (LO).
- Received signal, properly amplified, is mixed with LO to create beat.
- The frequency of beat signal is proportional to the distance between detected object and radar.

By including VCO device as LO, we can carry out different frequency modulations by changing voltage control waveform:

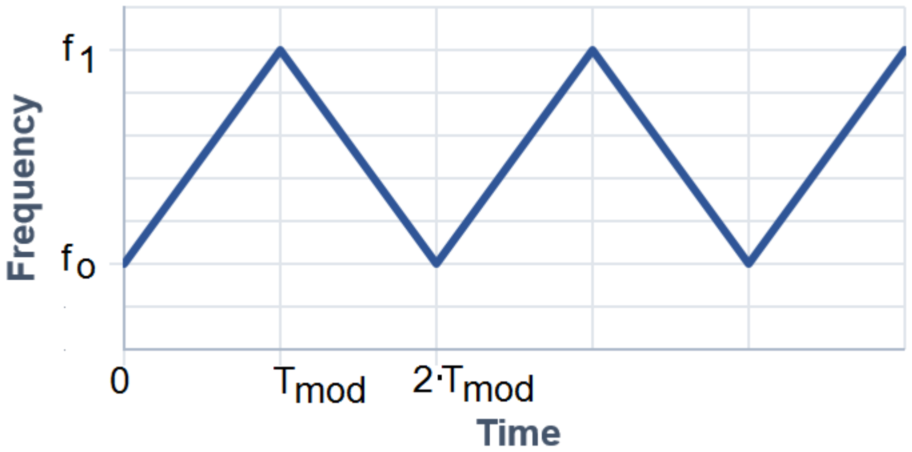


COST is supported by the
EU Framework Programme Horizon2020

Basic principles

Linear modulation

Transmitted chirp signal in time domain is a linear ramp in frequency domain.



$$f' = \frac{f_1 - f_o}{T_{mod}}$$

Transmitted signal is a frequency modulated sine wave

$$S_{Tx} = sin\left[2\pi t\left(f_o + \frac{f' \cdot t}{2}\right)\right]$$

COST is supported by the
EU Framework Programme Horizon2020

Basic principles
Linear modulation

Received signal is a delayed sine wave, again frequency modulated

$$S_{Rx} = sin\left[2\pi(t - t_d)\left(f_o + \frac{f' \cdot (t - t_d)}{2}\right)\right]$$

where delay correlates the object distance (*d*) and light velocity in the medium (*c*)

$$t_d = \frac{2d}{c}$$

When transmitted and received signals are mixed and filtered by means of low-pass filter, the resulting signal (S_{out}) shows a frequency (f_{out}) proportional to target distance

$$f_{out} = f' \cdot \ t_d$$

The maximum range of FMCW radar depends on frequency sweep width

$$\Delta R = \frac{c}{2 \cdot (f_1 - f_o)}$$

Basic principles

Signal processing

FFT

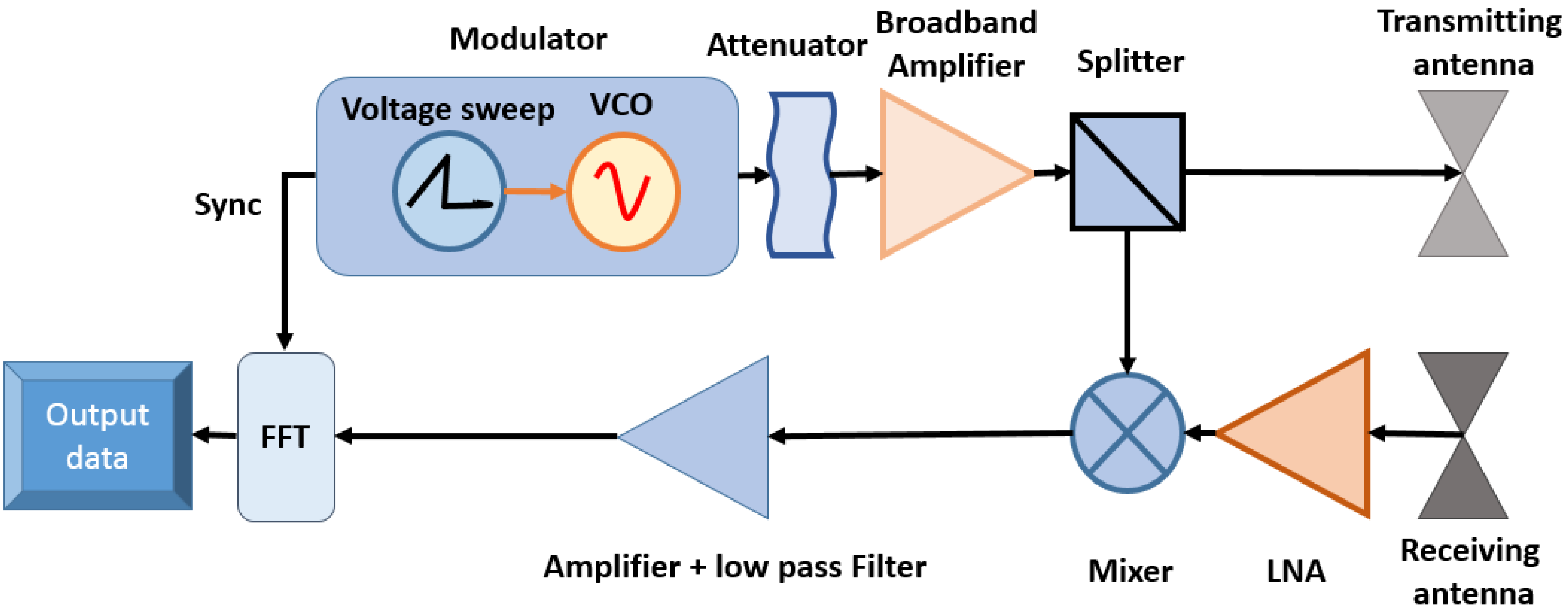
Applying Fast Fourier Transform to output signal allows to obtain frequency data, and consequently to calculate distance of multiple detected objects.

The use of low pass filter increases signal to noise ratio, decreasing broadband high frequency noise.

COST is supported by the
EU Framework Programme Horizon2020

Basic principles

Basic block diagram of FMCW radar



FFT and Output data blocks are carried out inside the PC

COST is supported by the
EU Framework Programme Horizon2020

Guide

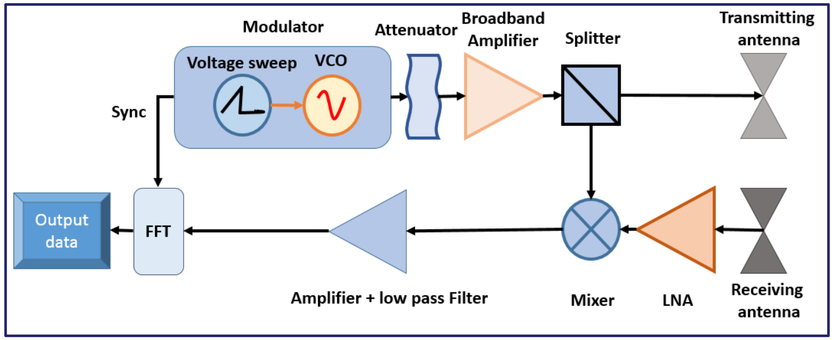
RF Hardware design

COST is supported by the
EU Framework Programme Horizon2020

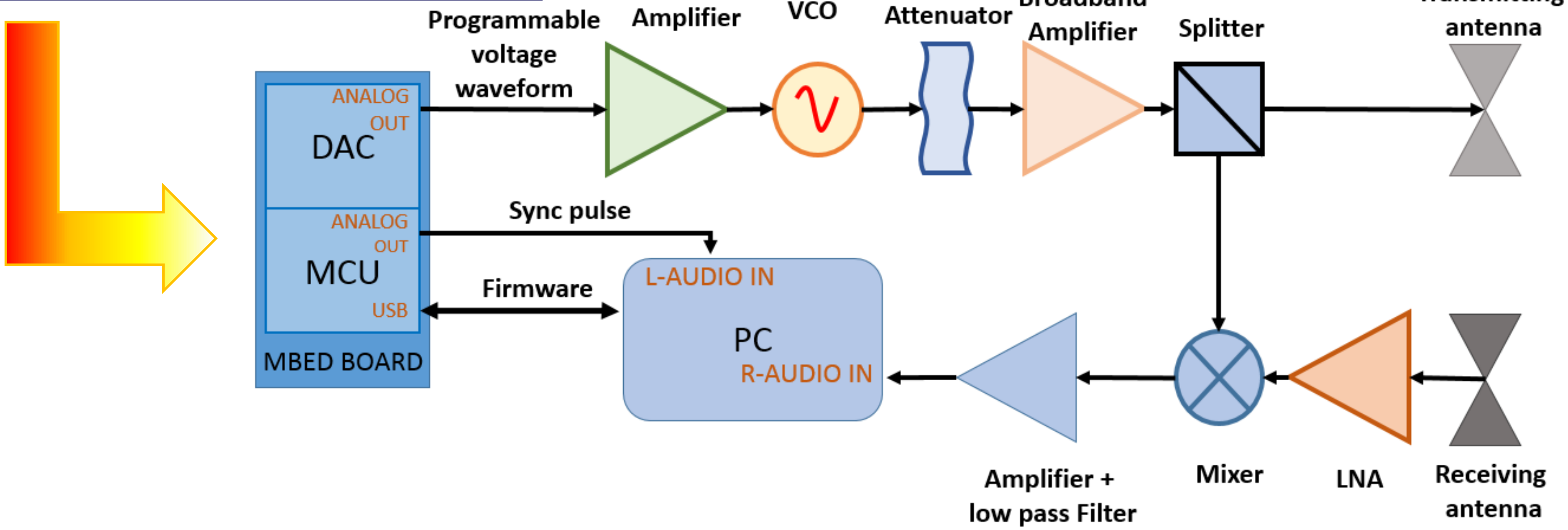
Guide

RF Hardware design

In order to increase interactive features and versatility of modulation shapes, we replaced analog function generator with a versatile mbed LPC1768 microcontroller (MCU) board, with high performance ARM Cortex-M3 core.



Block diagram of FMCW radar changed from basic to new version that adds control lines and connection to the PC.

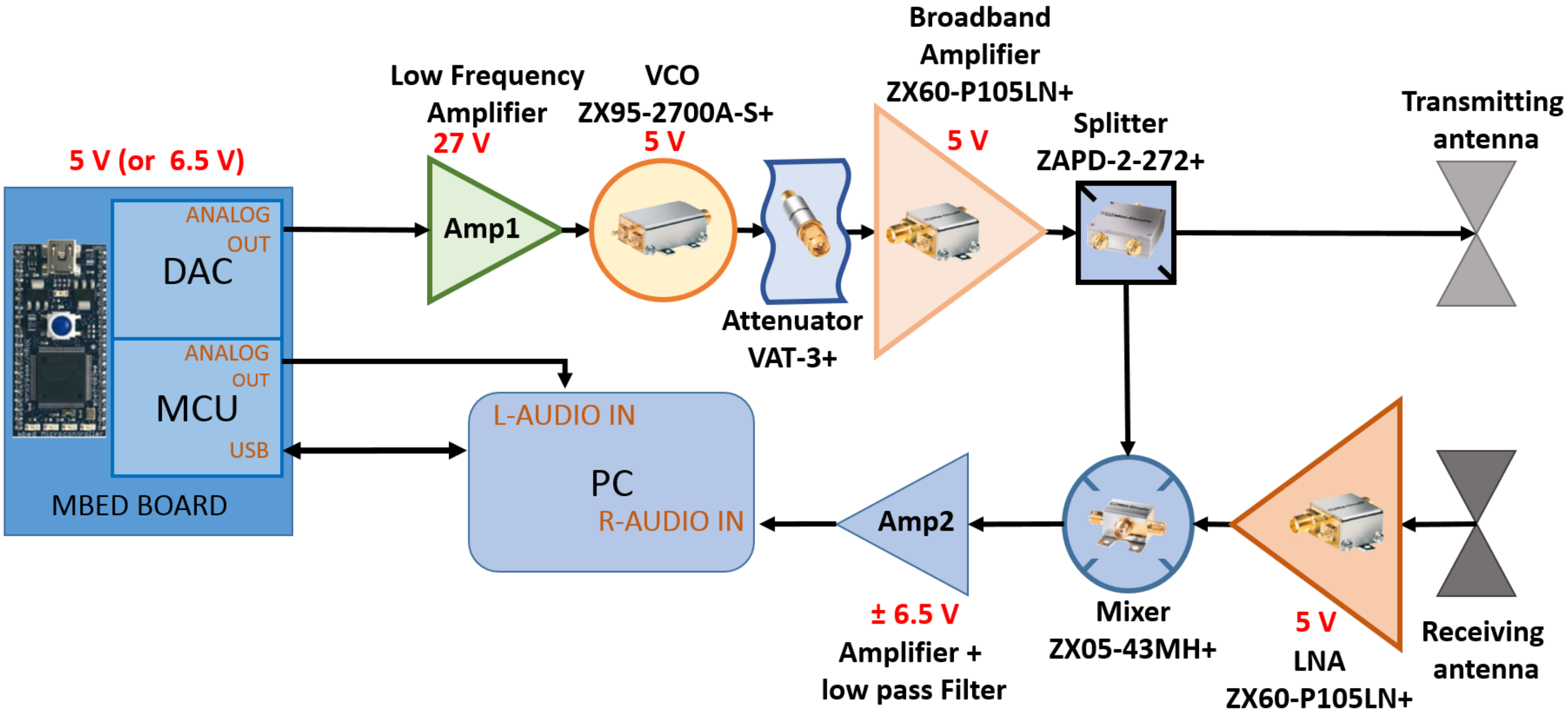


COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design – power supply requirements

Components and devices included in the project require DC voltages sources not always of equal value. Consequently, the design of the GPR must be complemented by the power system design in order to generate all necessary voltage levels. Moreover, to enable proper operation autonomy, a rechargeable 12V lithium battery will be inserted.



COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

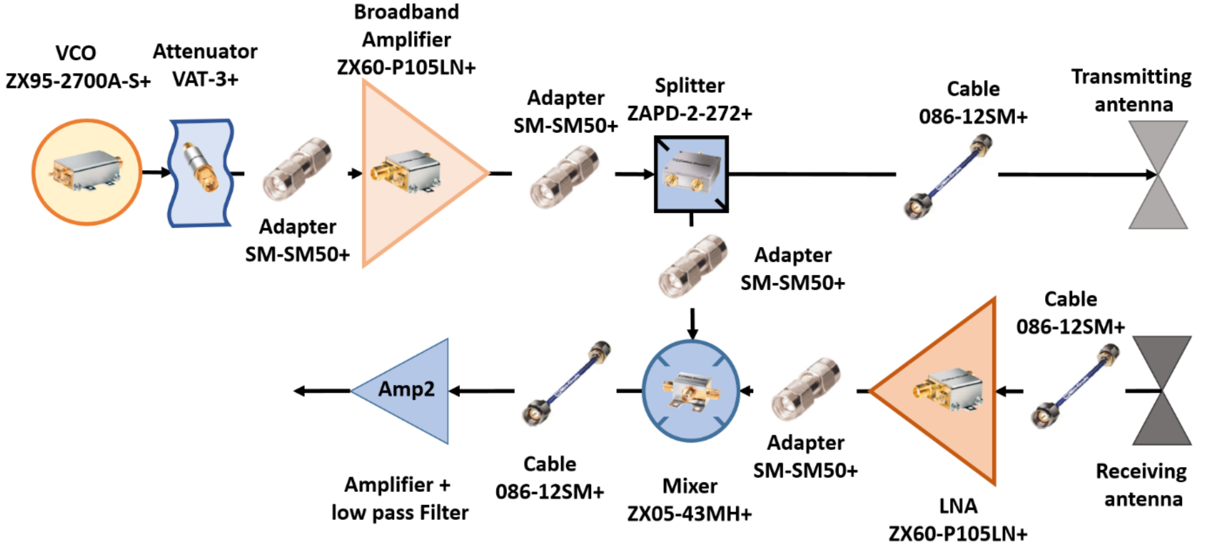
Bill of material – RF subsystem.

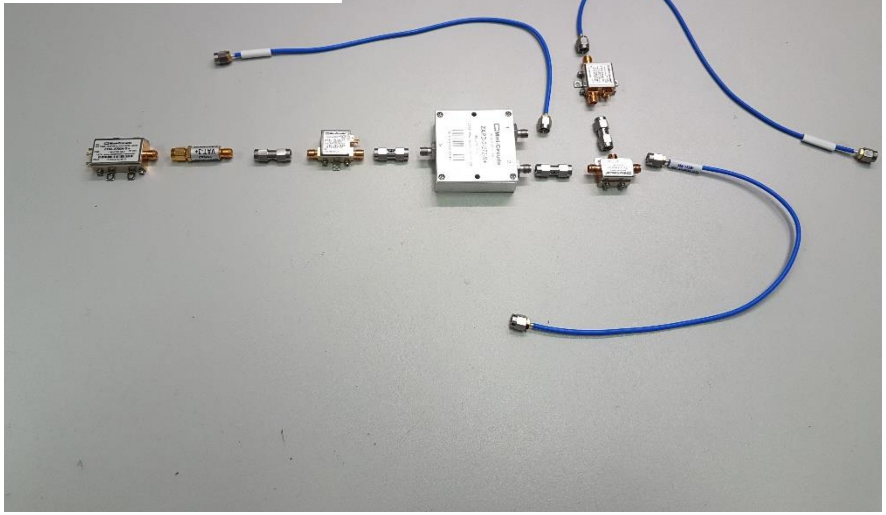
Part number	Qty	Description	Manufacturer	datasheet	Approximate cost (€)
ZX95-2700A-S+	1	Voltage Controlled Oscillator Wide Band 1300 to 2700 MHz	Mini-Circuits	ZX95-2700A+.pdf	54.36
VAT-3+	1	SMA Fixed Attenuator 50Ω 1W 3dB DC to 6000 MHz	Mini-Circuits	VAT-3+.pdf	13.95
ZX60-P105LN+	2	Low Noise Amplifier 50Ω 40 to 2600 MHz	Mini-Circuits	ZX60-P105LN+.pdf	76.13 x 2
ZAPD-2-272-S+	1	Power Splitter/Combiner 2 Way-0° 50Ω 800 to 2700 MHz	Mini-Circuits	ZAPD-2-272+.pdf	81.57
ZX05-43MH-S+	1	Frequency Mixer wide band Level 13 (LO Power +13 dBm) 824 to 4200 MHz	Mini-Circuits	ZX05-43MH+.pdf	46.45
SM-SM50+	4	Adapter, SMA-M to SMA-M 50Ω DC to 18 GHz	Mini-Circuits	SM-SM50+.pdf	13.95 x 4
086-12SM+	3	Coaxial Cable 50Ω DC to 18 GHz	Mini-Circuits	086-12SM+.pdf	11.92 x 3

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design – assembly of the components

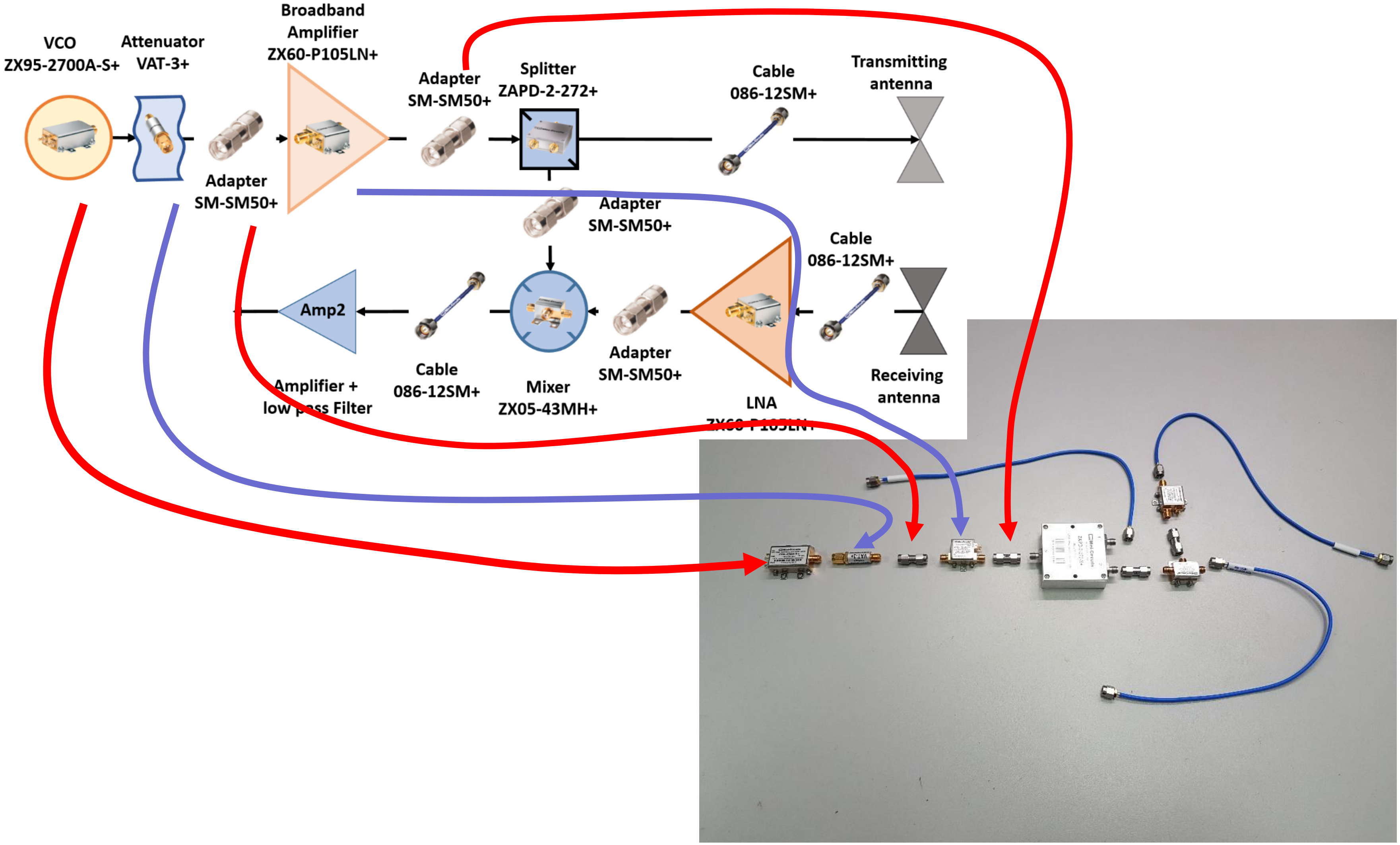




COST is supported by the
EU Framework Programme Horizon2020

Guide

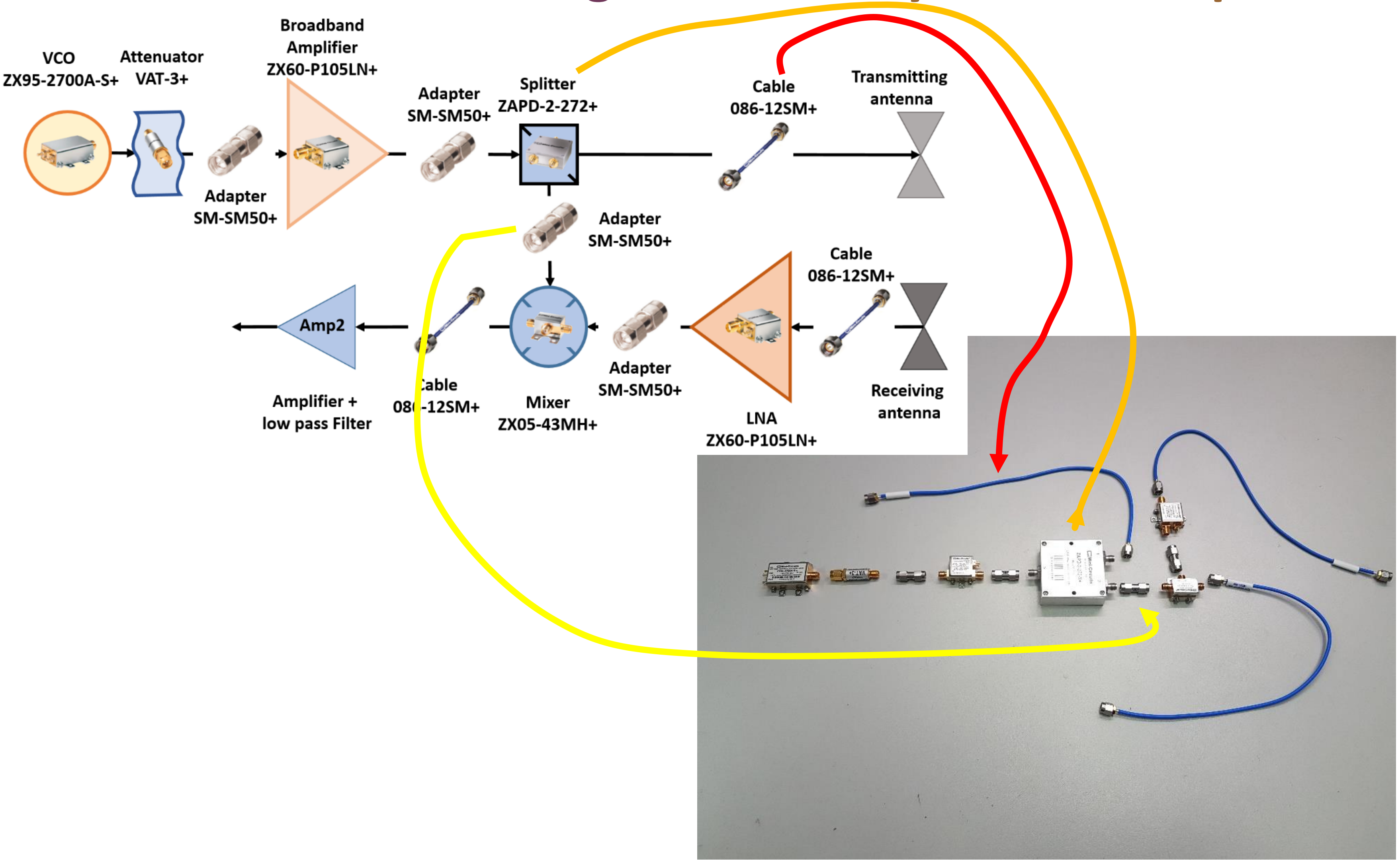
RF Hardware design – assembly of the components



COST is supported by the
EU Framework Programme Horizon2020

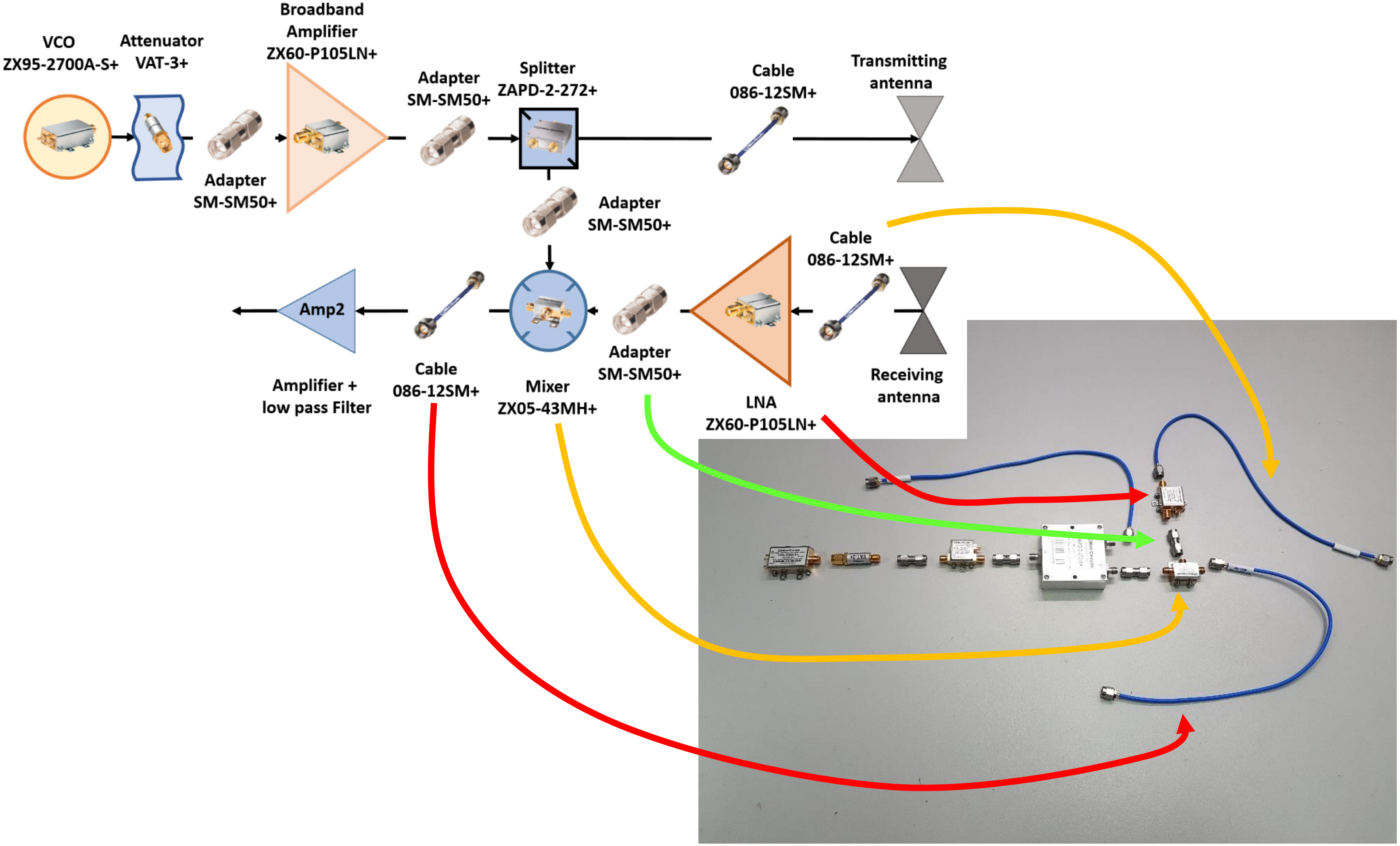
Guide

RF Hardware design – assembly of the components



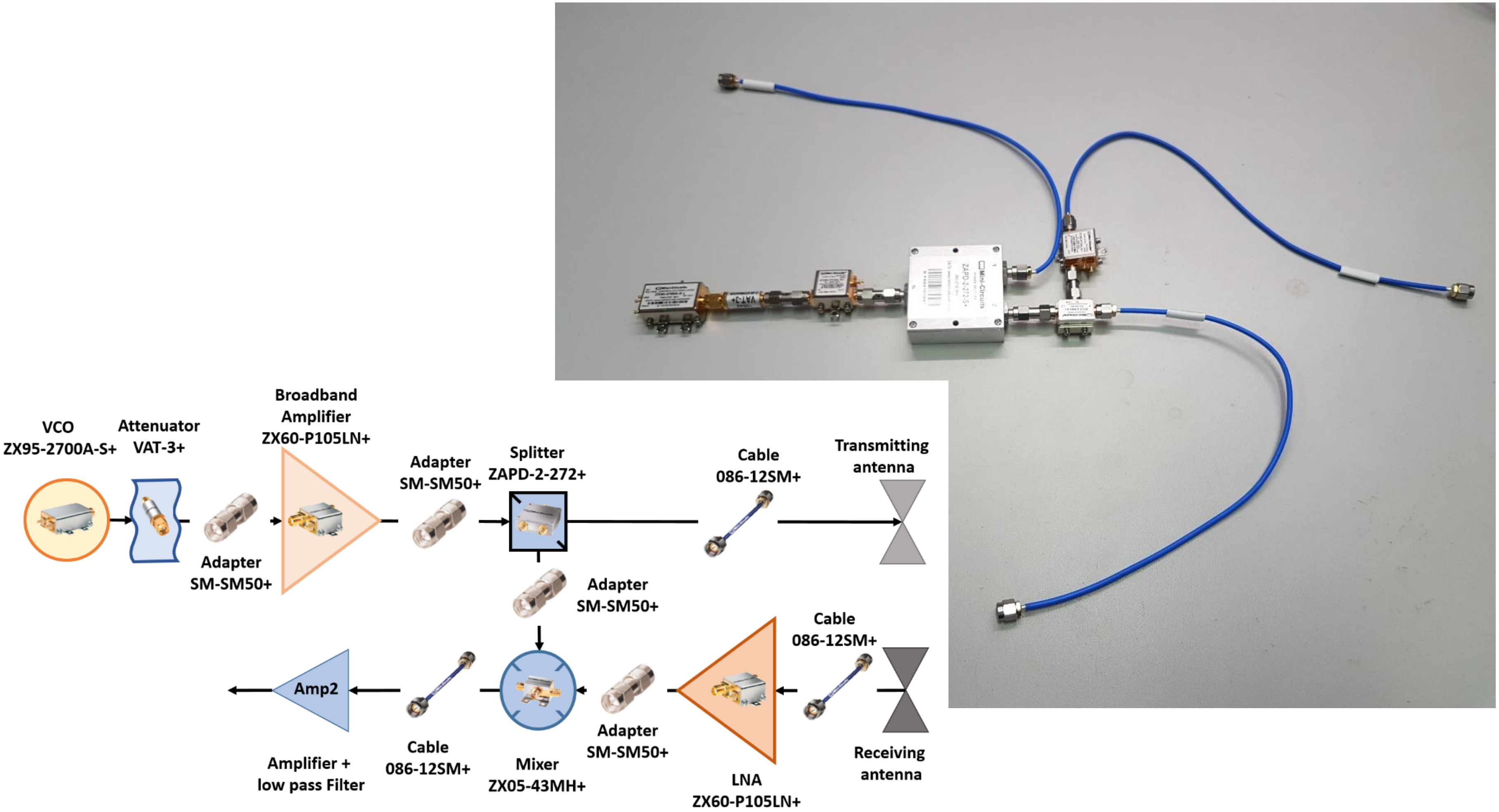
Guide

RF Hardware design – assembly of the components



Guide

RF Hardware design – assembly of the components



COST is supported by the
EU Framework Programme Horizon2020

RF Hardware design – PCB specification

The remaining sections concerning electronic circuits built by using of PCB (printed circuit board). In order to simplify realization of PCBs, we used single layer PCBs.

Working in our laboratory, equipped with specific instrumentation, we used the following temporization for printed circuit board realization:

- ❑ After the creation of the PCB layout by software tool for designing circuit board and a laser printer, the photoresist have to be placed on top of the copper layer for covering it.
- ❑ Using a mask aligner device the positive photoresist film (like AZ1518), deposited on the layer of copper, must be impressed considering a time of 16 seconds.
- ❑ After that, start the operation of soft bake for a time of develop of one minute with a mix AZ 351B developer and deionized water by volume as 1 part of developer and 4 parts of water for obtaining a good contrast, at 100°C of temperature
- ❑ With the help of a microscope is possible to check if all the connection is correct.
- ❑ In the *hard bake* phase the board is cooked at 115°C with a HT-302D ATV-Technology for a time of 2 min.
- ❑ The final phase is a bath in a solution of ferric chloride for a time of around 10 minutes.
- ❑ The board have to be cleaned with acetone and is possible to do tin-plating adding a layer of tin to help the next operation of weld the component.

But the feature single layer of PCB allows it to be realized in non-specialized laboratory, with few components.

Guide

RF Hardware design

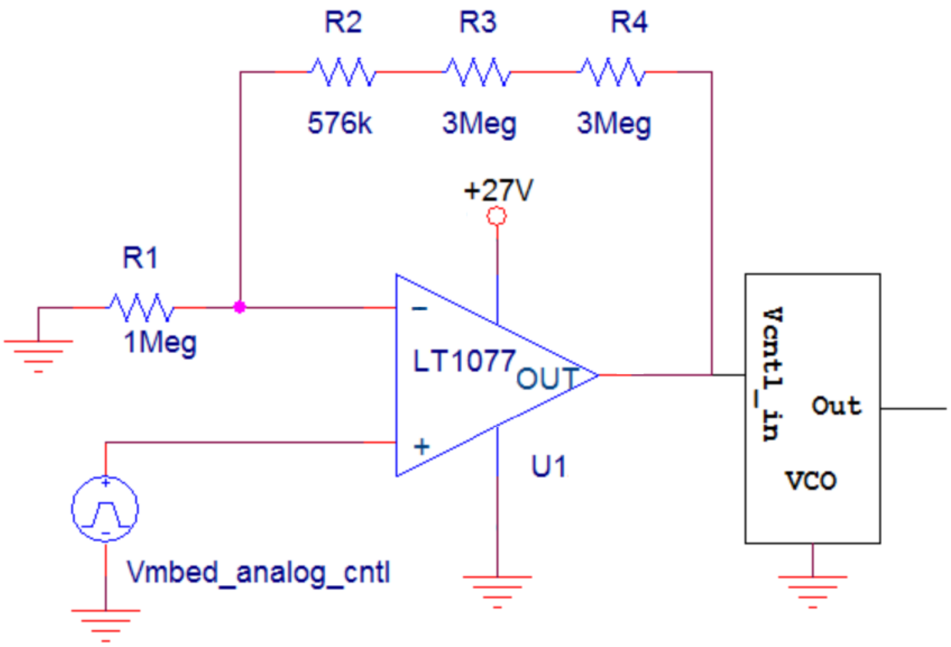
Bill of material – Low frequency amplifier subsystem (Amp1).

Part number	Qty	Description	Manufacturer	datasheet	Approximate cost (€)
LT1077S8#PBF	1	Micropower, Single Supply, Precision Op Amp (U1)	Linear Technology	LT1077fa.pdf	2.61
WCR0805-1M0FI	1	Resistor chip SMD, 1 Meg, 150 V, 0805 [2012 Metrico], 125 mW, ± 1% (R1)	TE Connectivity	-	0.0383
ERJ3EKF5763V	1	Resistor SMD, 576kΩ ±1%, 0,1W, ± 1% (R2)	Panasonic	-	0.014
CRCW08053M00F KEA	2	Resistor SMD, 3MΩ, ±1%, 0.125W (R3, R4)	Vishay	-	0.016 x 2
282834-2	3	2 Position Wire to Board Terminal Block Horizontal with Board 0.100" (2.54mm) Through Hole (P1,P2,P3)	TE Connectivity AMP Connectors	-	0.8

Guide

RF Hardware design

Low frequency amplifier (Amp1) subsystem- Schematics.



Values of the resistors result in an amplification of the non-inverting stage equal to

Av = 7.576

A signal *Vmbed_analog_out* that shows an amplitude range of 0÷3.3V is amplified in a correspondent amplitude interval 0÷25V of the output signal (*Vcntl_in*).

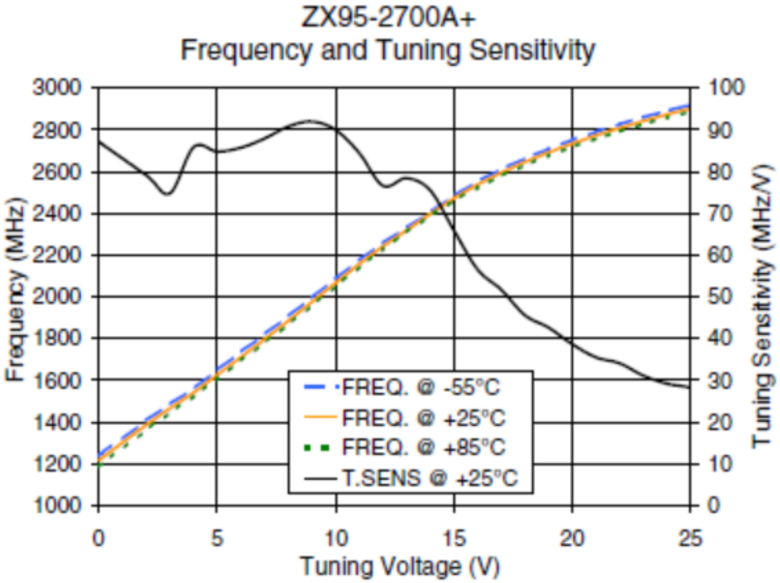
This interval is coincident with the control voltage maximum range of VCO, and allows maximum range of frequency shift.

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Low frequency amplifier (Amp1) subsystem- some considerations.



In the case of triangular FM, the nonlinear characteristic <<tuning voltage-frequency>> of the VCO imposes the use of a control voltage waveform different that triangular for compensating the compression showing at higher voltage values.

In any case, we are able to generate a maximum frequency equal to 2902.5 MHz (@25°C)

Unfortunately the choose of broadband amplifier ZX60-P105LN+ has been a trade-off between performance and cost. So, the upper frequency limit of bandwidth of amplifier determines maximum value of allowed frequency modulation.

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Low frequency amplifier (Amp1) subsystem- some considerations.

If we continue to use this broadband amplifier, two alternatives are possible:

- ❖ To impose 2600 MHz as maximum frequency that an operator could select in the input data mask. The disadvantage is the use of a lower dynamic for analog output of the mbed, i.e. 2.24 V instead of the maximum value 3.3 V.
- ❖ For increasing data resolution, adjust the amplification value of amplifier, i.e. Av=5.15. This is result in necessary selection of new values of the resistances. And changes in the firmware are inevitable.

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Low frequency amplifier (Amp1) subsystem- some considerations.

Furthermore …

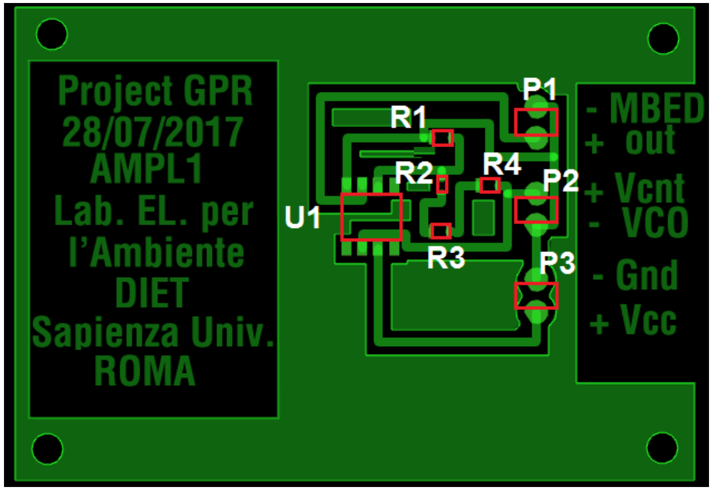
❑ The minimum value of the usable triangular wave period is about 0.625 ms. When the frequency of the same triangular wave exceeds 1.6kHz the gain of amplifier decreases progressively and control voltage value of VCO is wrong.

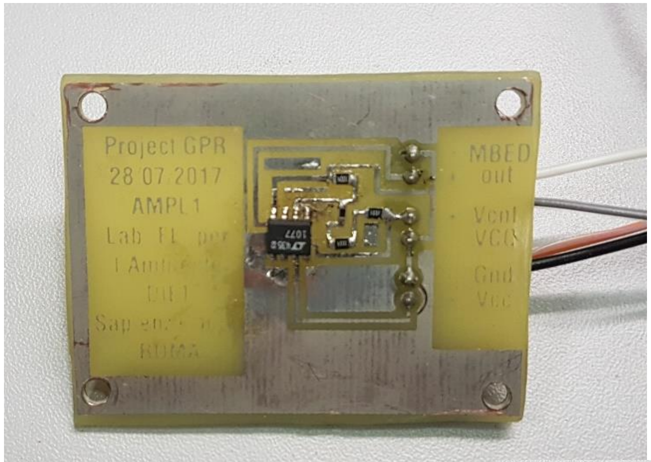
❑ If square waves or stepped signals are generated, consider SR (Slew Rate) value of the same amplifier, defined in the range 0.07 ÷ 0.12 V/ms, which limits the range choice of step width.

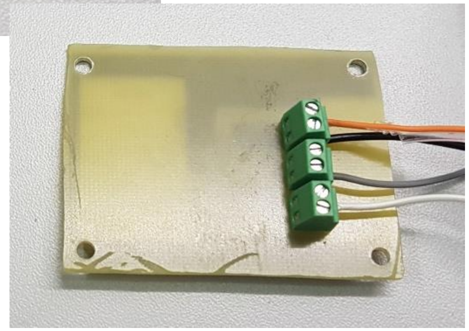
Guide

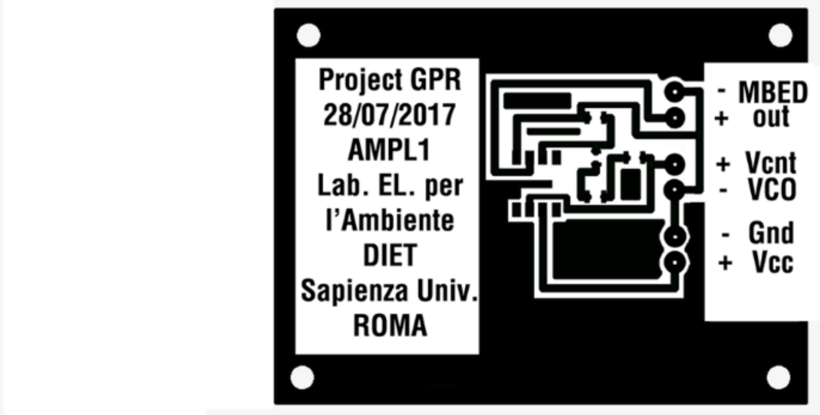
RF Hardware design

Low frequency amplifier (Amp1) subsystem – PCB design.









PCB mask file (scale 1:1) is:
Amp1_1_1.png

Guide

RF Hardware design

Bill of material – Amplifier+ Low pass filter (Amp2) subsystem.

Part number	Qty	Description	Manufacturer	datasheet	Approximate cost (€)
OP467	1	Quad Precision, High Speed Operational Amplifier IC OPAMP GP 28MHZ 14DIP (OP467GP)	Analog Device	OP467.pdf	16.04
CCF0710K0JKE36	1	RES 1/4W 10K 5% (R1)	Vishay	-	0.085
RN55D8451FB14	1	RES 8.45K 1/8W 1% AXIAL (R4)	Vishay	-	0.1
MFR-25FBF52-7K15	1	RES 7.15K 1/8W 1% AXIAL (R5)	Vishay	-	0.1
YR1B102KCC	1	RES 102K 1/4W 1% AXIAL (R6)	TE Connectivity	-	0.286
RN60D1212FB14	1	RES 12.1K OHM 1/4W 1% AXIAL (R8)	Vishay	-	0.21
RN55C1001FB14	2	RES 1K 1/8W 1% AXIAL (R7, R13)	Vishay	-	0.11
RN55D1742FB14	1	RES 17.4K 1/4W 1% AXIAL (R9)	Vishay	-	0.96
MFR-25FBF52-28K	1	RES 28K 1/4W 1% AXIAL (R10)	Yageo	-	0.08
MFR-25FBF52-4K12	1	RES 4.12K 1/4W 1% AXIAL (R11)	Yageo	-	0.08

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Bill of material – Amplifier+ Low pass filter (Amp2) subsystem. - Continue

Part number	Qty	Description	Manufacturer	datasheet	Approximate cost (€)
RN55D1621FB14	1	RES 1/8W 1.62K 1% (R12)	Vishay	-	0.1
RN55D1100FRE6	2	RES 1/8W 110ohms 1% 100ppm (R2,R3)	Vishay / Dale	-	0.094
RN55D1100FRE6	2	RES 1/8W 110ohms 1% 100ppm (R2,R3)	Vishay / Dale	-	0.094
CFR16J47K	1	RES 47K 1/4W 1200PPM (R14)	TE Connectivity / Neohm	-	0.085
489D105X0025A6VE3	1	CAP 1µF ±20% 25V c.c., +85°C (C1)	Vishay	-	0.678
A102J15C0GF5TAA	4	CAP 1000pF 50V 5% C0G Axial (C2,C3,C4,C5)	Vishay / BC Components	-	0.281
PV36Y103C01B00	1	Trimmer resistivo Murata serie PV36W 10kΩ, ±10%, 0,5W, 25 turn (Trimmer)	Murata	-	1.37

Guide

RF Hardware design

Bill of material – Amplifier+ Low pass filter (Amp2) subsystem. - Continue

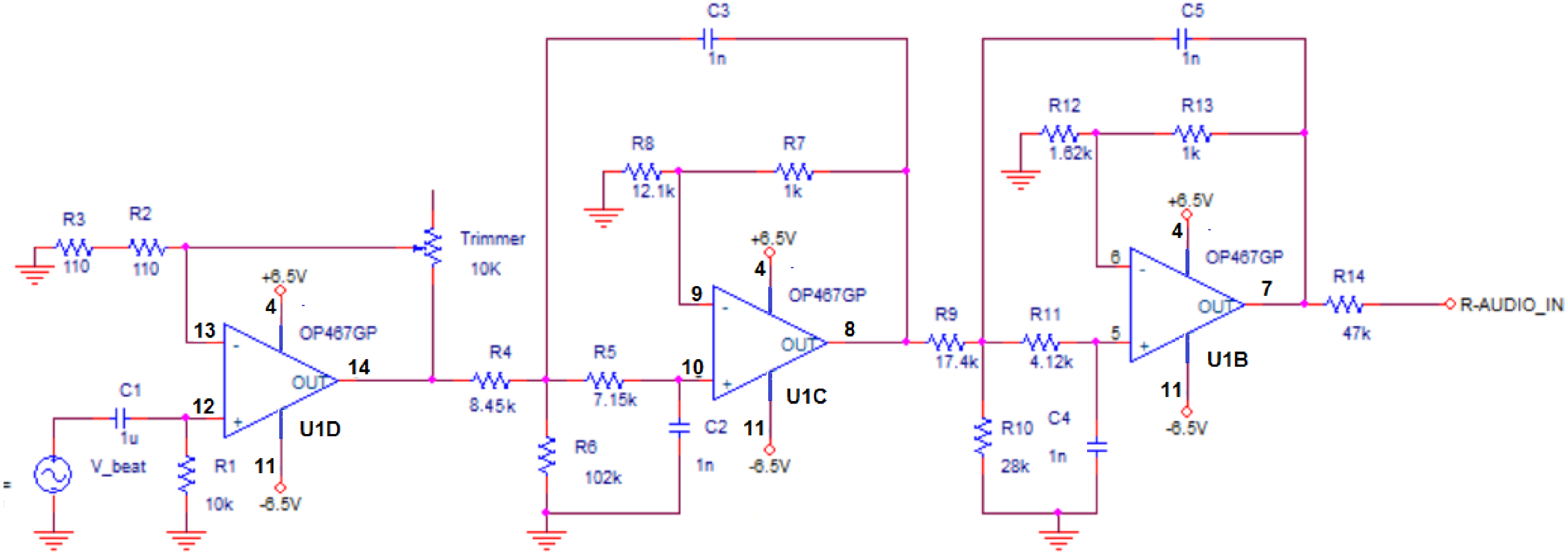
Part number	Qty	Description	Manufacturer	datasheet	Approximate cost (€)
901-9889-RFX	1	Connector, rf coaxial, sma blkhd ecept, front, mount ,solder cup terminal, nickel plt [V_beat(J1)]	Amphenol RF	-	9.12
282834-3	1	3 Position Wire to Board Terminal Block Horizontal with Board 0.100" (2.54mm) Through Hole (P1)	TE Connectivity AMP Connectors	-	1.64
282834-2	1	2 Position Wire to Board Terminal Block Horizontal with Board 0.100" (2.54mm) Through Hole (P2)	TE Connectivity AMP Connectors	-	0.8

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Amplifier+ Low pass filter (Amp2) subsystem[4]. Schematics



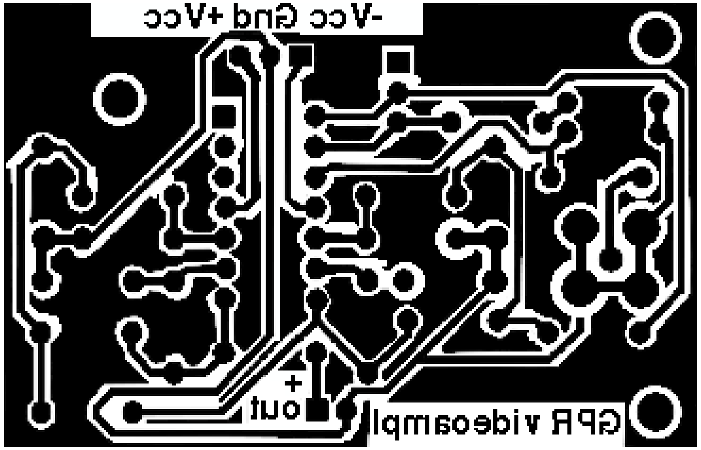
The amplification is set by the trimmer and the filtering modules determine the maximum bandwidth frequency around 15 kHz..

COST is supported by the
EU Framework Programme Horizon2020

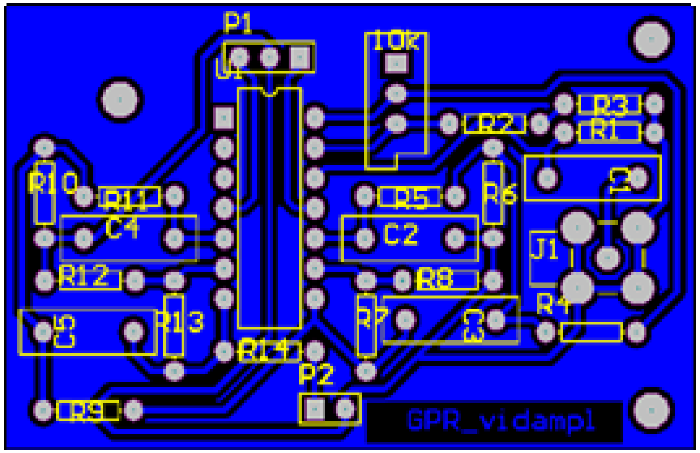
Guide

RF Hardware design

Amplifier+ Low pass filter (Amp2) subsystem. PCB design



PCB mask file (scale 1:1) is: PCB videoampl_1_1.png



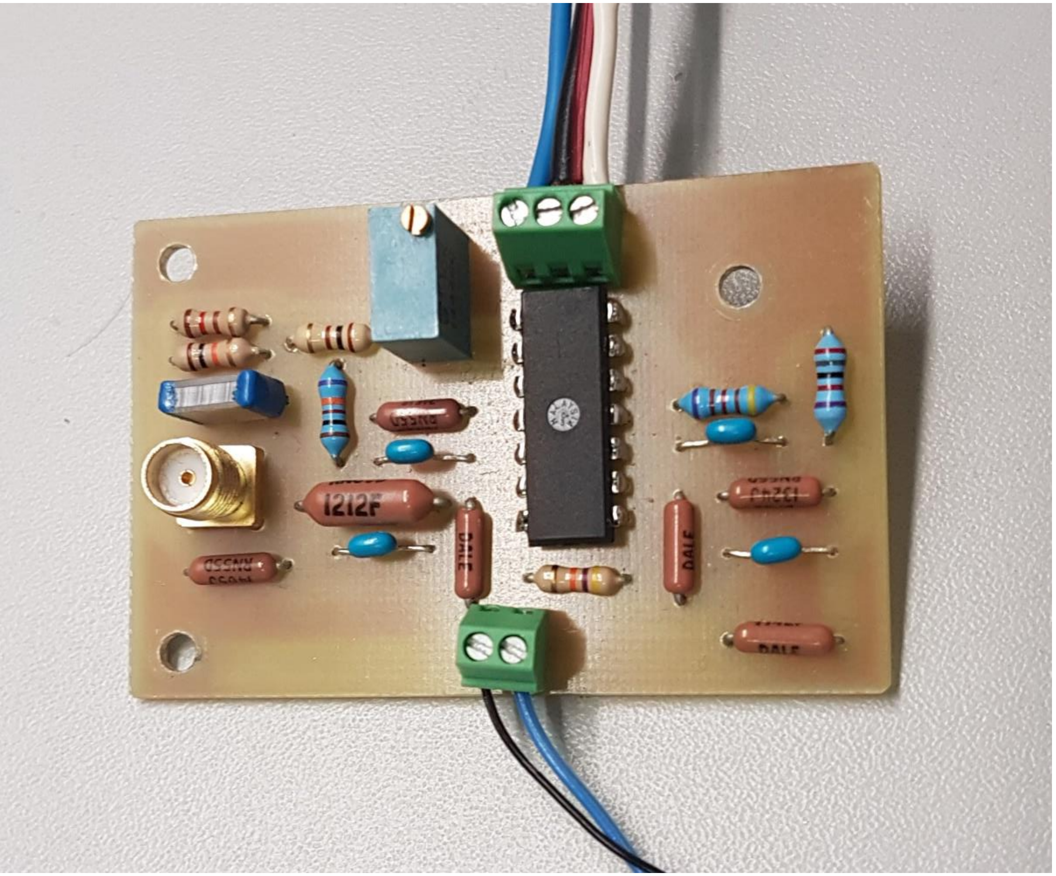
Connect «+ out» pin of connector P2 by means of a wire to the right channel of the laptop audio input. The same connector jack used for the mbed board («+ Sync pulse» signal).

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Amplifier+ Low pass filter (Amp2) subsystem. PCB design

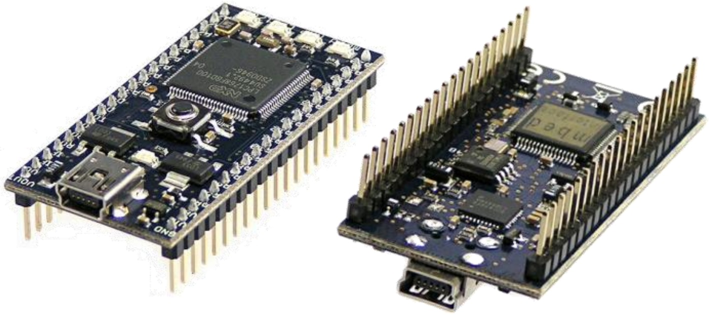


Guide

RF Hardware design

Bill of material – mbed board

Part number	Qty	Description	Manufacturer	datasheet	Approximate cost (€)
MBED-LPC1768	1	ARM mbed LPC1768 prototyping board (MBED)	mbed	LPC1768.pdf, mbed-005.1.pdf, lpc17xx_um.pdf, lpc17xx_ds.pdf, pc17xx_es.pdf,	58.23
282834-2	3	2 Position Wire to Board Terminal Block Horizontal with Board 0.100" (2.54mm) Through Hole (P1,P2,P3)	TE Connectivity AMP Connectors	-	0.8 x3
801-43-020-10-012000	2	20P single row (female) lo profile SKT long ins	Mill-Max	-	1.86 x 2



MBED-LPC1768

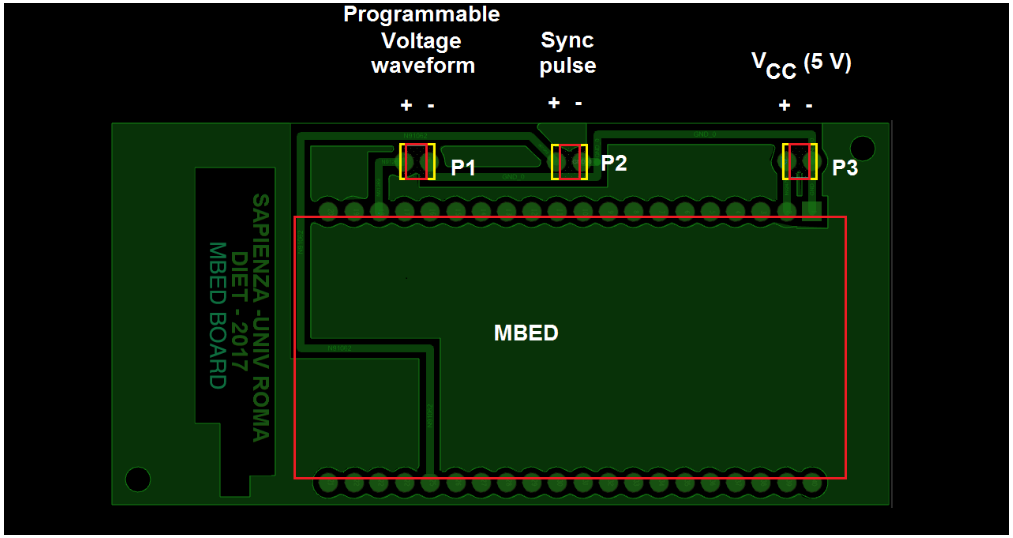
Guide

RF Hardware design

mbed board - PCB design

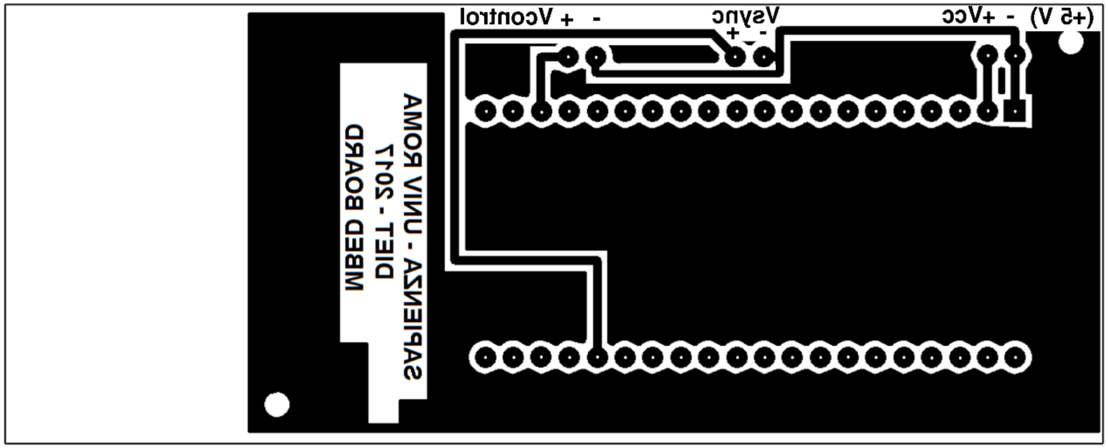
Connect «+ Sync pulse» wire to left channel of the laptop audio input by means of connector jack. The same connector jack used for the video amplifier (Amp2) output signal.

Pads signed "–" coincide with ground, and you could connect only one of these.



Vcc should be selected in the range 4.5V÷9 V.

You can select one of the two +5V or +6.5 V voltages generated by DC-DC converter (circuit No. 1).



PCB mask file (scale 1:1) is: mbed_1_1.png

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Bill of material – Power supply subsystem (DC-DC converters).

Part number	Qty	Description	Manufacturer	datasheet	Approx. cost (€)
TSRN 1-2465	2	DC/DC converter 6.5V 1° (U2,U3)	Traco Power	TSRN-1 DC_DC converter.pdf	12.08 x 2
R-78E5.0-0.5	1	CONV DC/DC 5V 500MA OUT THRU (U1)	Recom Power	R-78Exx-0.5 DC_DC converter.pdf	2.24
282834-4	2	4 Position Wire to Board Terminal Block Horizontal with Board 0.100" (2.54mm) Through Hole (P1, VinDC+VnSHDN+Gnd+Vout)	Buchanan - TE Connectivity	-	1.67 x 2
447687 (supplier: Jonathan ®)	1	FullPower - Battery Lipo 4S 2200mAh 35C Silver V2 – DEANS (BT1)	Full Power	-	25,90
C0603X103F5JACTU	2	Multilayer Ceramic Capacitor (MLCC) Standard 10000pF, ± 5%, 50 V DC, SMD (C1,C2)	KEMET	-	0.52 x 2
282834-2	1	2 Position Wire to Board Terminal Block Horizontal with Board 0.100" (2.54mm) Through Hole (BT1)	TE Connectivity AMP	-	0.8

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Bill of material – Power supply subsystem (DC-DC converters) - Continue

Part number	Qty	Description	Manufacturer	datasheet	Approx. cost (€)
LT3461AES6#TRMPBF	1	(Step Up) DC-DC, Reg. 2.5V-16Vin, 1.25V-38Vout, 250mAout, TSOT-23-5	Linear Technology	LT3461Afa.pdf	1.89
GRM21BR71H105KA12L	1	CAP (MLCC) Murata GRM 1µF, ±10%, 50 V c.c., SMD (C1_2)	Murata	-	0.126
C1206C474K5RACTU	1	CAP (MLCC) KEMET C 470nF, ±10%, 50 V, SMD (C2_2)	Kemet	-	0.166
C1206C220J5GACTU	1	CAP (MLCC) KEMET C 22pF, ±5%, 50 V c.c., SMD (C3)	Kemet	-	0.074
MCTC0525B4533T5E	1	RES 453 kohm, 100 V, 100 mW (R1)	Multicomp	-	0.447
ERA6ARB2212V	1	RES 22.1 kohm, 100 V, 0805, 125 mW (R2)	Panasonic	-	0.613
LQH66SN330M03L	1	Inductor 33 µH, 860 mA, 2525 (L1)	Murata	-	2.99

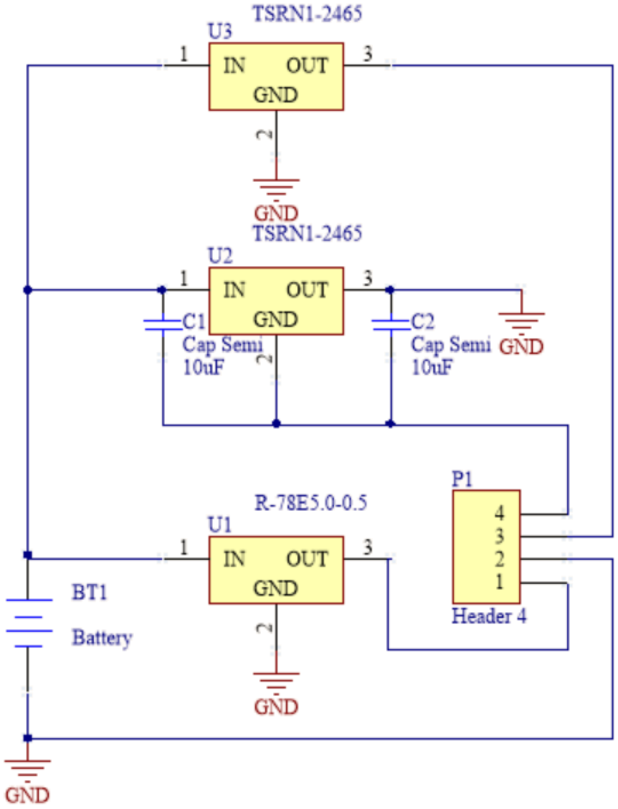
COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Power supply subsystem (DC-DC converters). Schematic of circuit No.1.

Scheme of circuit n.1 for converting
from 12V DC to four levels of output DC voltages : 5 V, ± 6.5V, and GND



❑ BT1: Full Power© Li-Po 4S 2200 mAh 35C battery

❑ U1,U2,U3: Efficiency of switching regulators: 91%÷93%

U2,U3

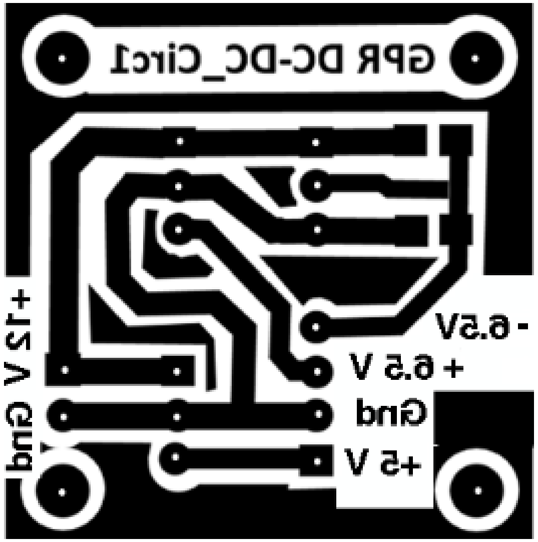
U1

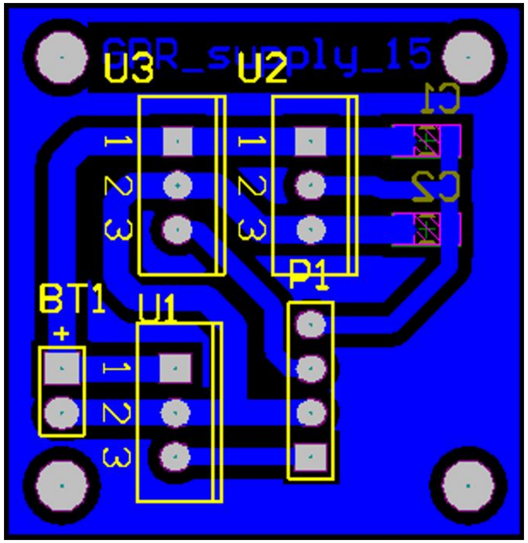
COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

Power supply subsystem (DC-DC converters). PCB design of circuit 1





PCB mask file (scale 1:1) is:
DC_1_PCBbn_1_1.png

C1 and C2 mounted on the bottom layer, the solder layer.

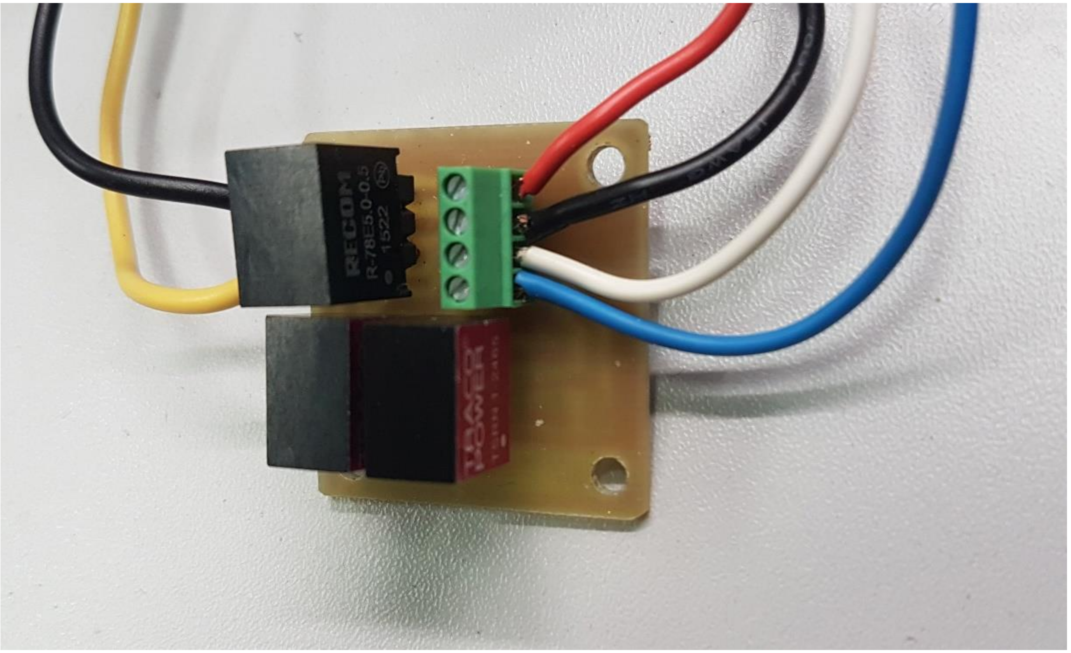
Warning! We recommend not welding directly to the copper layer the TSRN 1-2465 converters that do not tolerate high temperature.
Use 3P single row (female) socket connector SKT.

COST is supported by the
EU Framework Programme Horizon2020

Guide

RF Hardware design

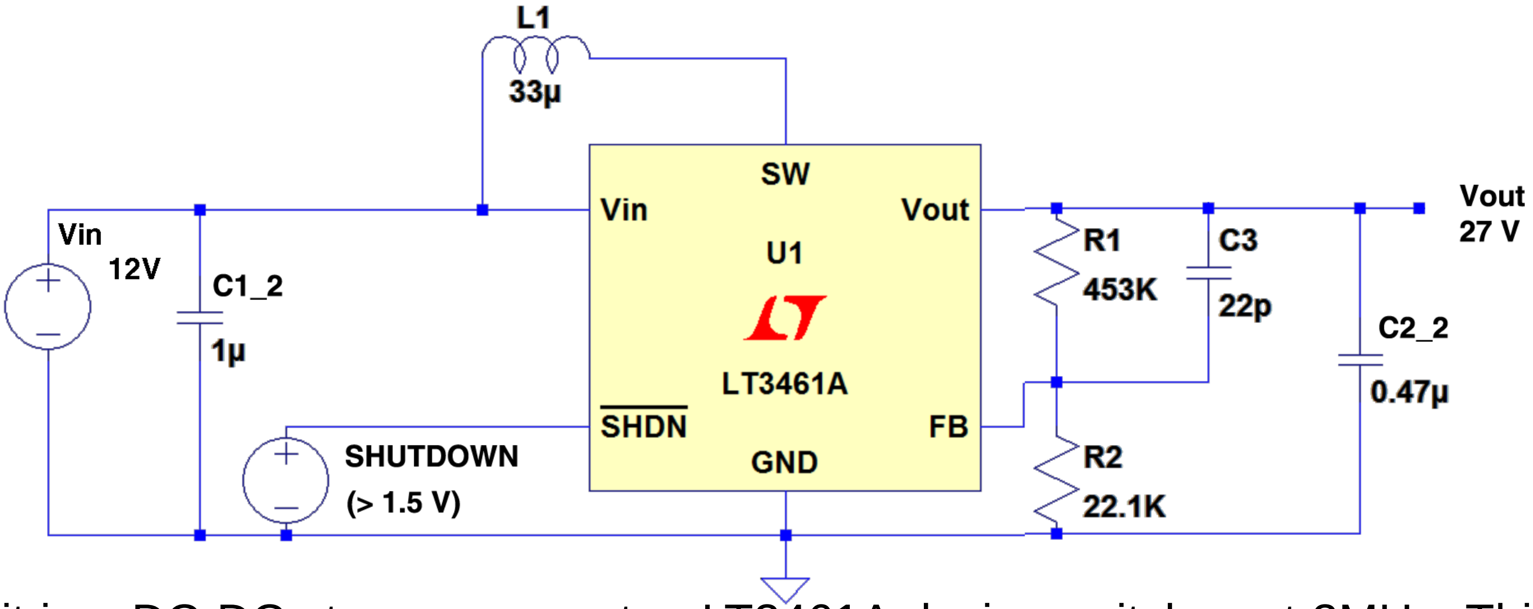
Power supply subsystem (DC-DC converters). PCB design of circuit 1



Guide
RF Hardware design

Power supply subsystem (DC-DC converters). Schematic of circuit No.2.

Scheme of circuit n.2 for converting from 12V DC to 27 V



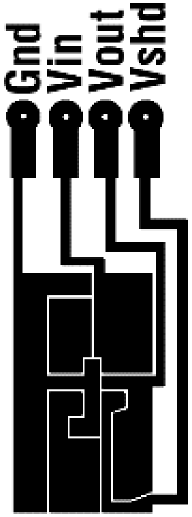
Circuit is a DC-DC step-up converter. LT3461A device switches at 3MHz. This high switching frequency enables the use of tiny, low cost and low height capacitors and inductors.

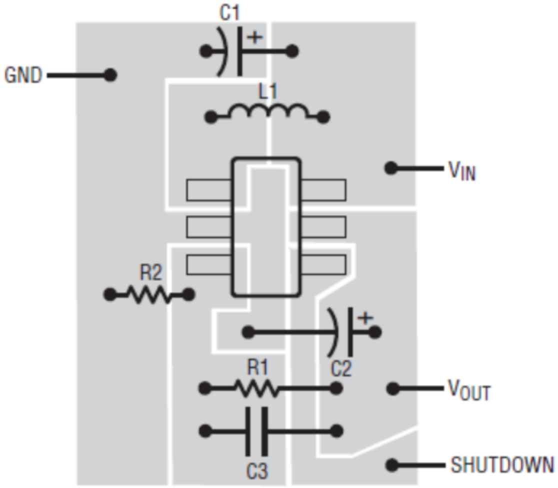
COST is supported by the
EU Framework Programme Horizon2020

Guide

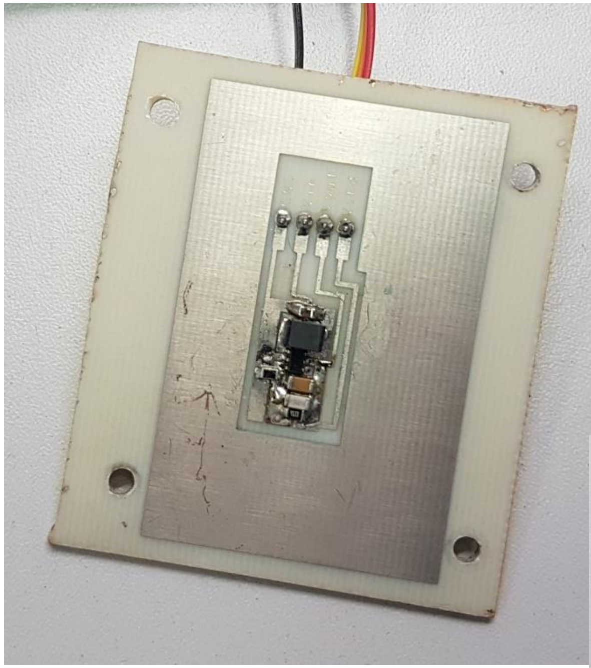
RF Hardware design

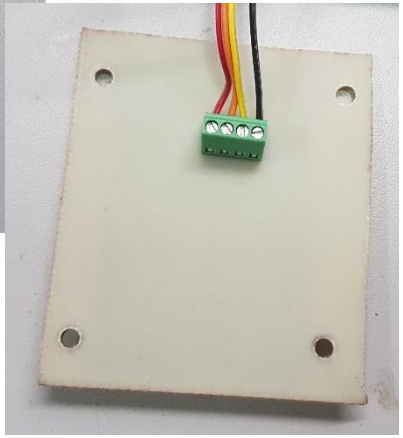
Power supply subsystem (DC-DC converters). PCB design of circuit 2





Suggested layout by Linear Technology.





PCB mask file (scale 1:1) is: DC_2_PCB_1_1.png

Guide

RF Hardware design

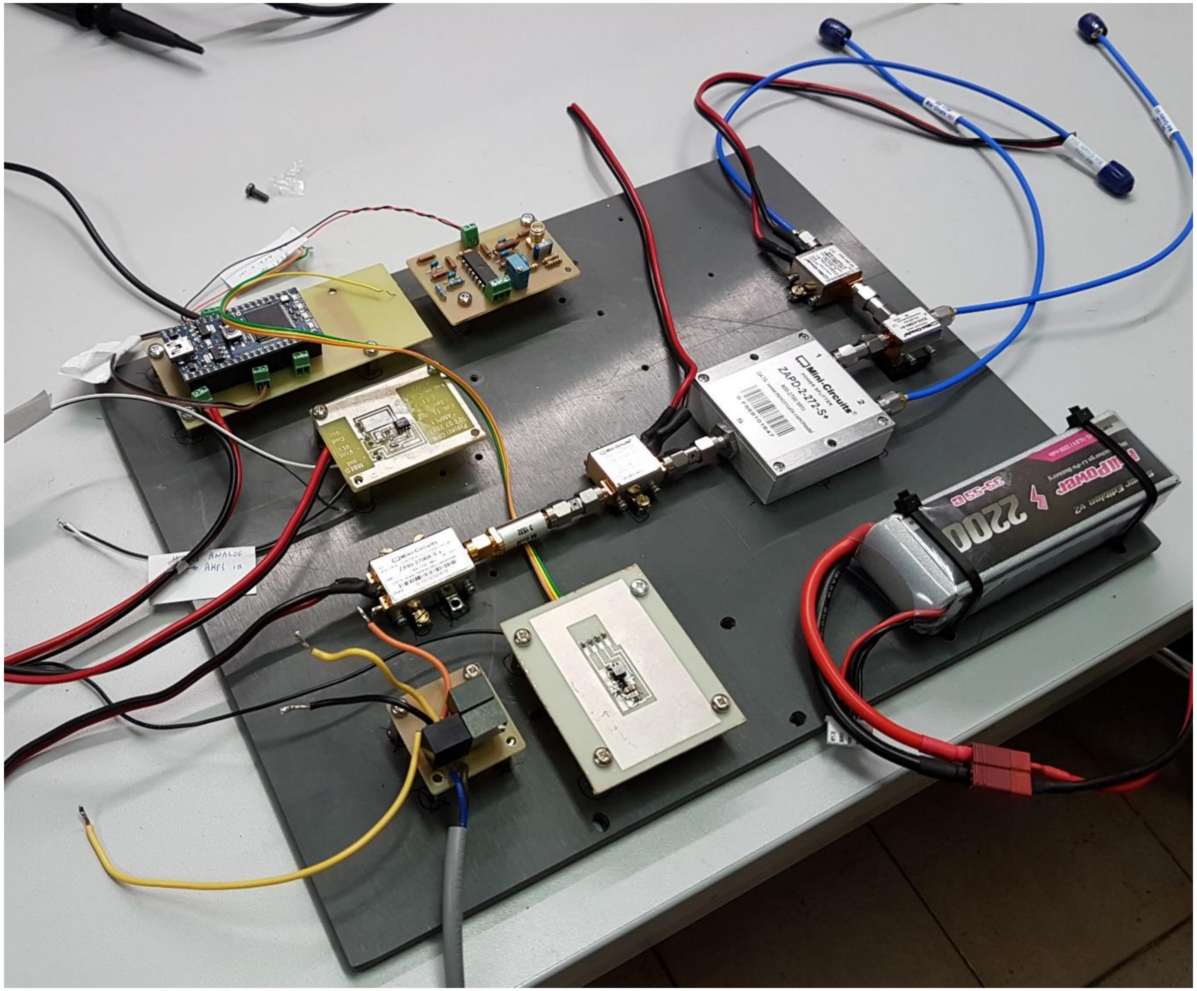
Bill of material – Miscellanea

Part number	Qty	Description	Manufacturer	Approximate cost (€)
1532 02	1	Audio stereo connector jack, 3.5 mm, male (R-AUDIO_IN)	Lumberg	2.61
AD10	1	Copper AD22 single layer 150 x 100 x 1.6mm FR4 copper sheet with 35µm copper thickness	CIF	2.92
	1	SMD291NL No Clean Tack Flux 5cc Syringe w / handle & Tip For Lead-Free Rework	CHIP QUICK	11.72
	as necessary	Solder Wires	MULTICOMP	
	as necessary	Screws		
	as necessary	Electrical wires for the connections		

COST is supported by the
EU Framework Programme Horizon2020

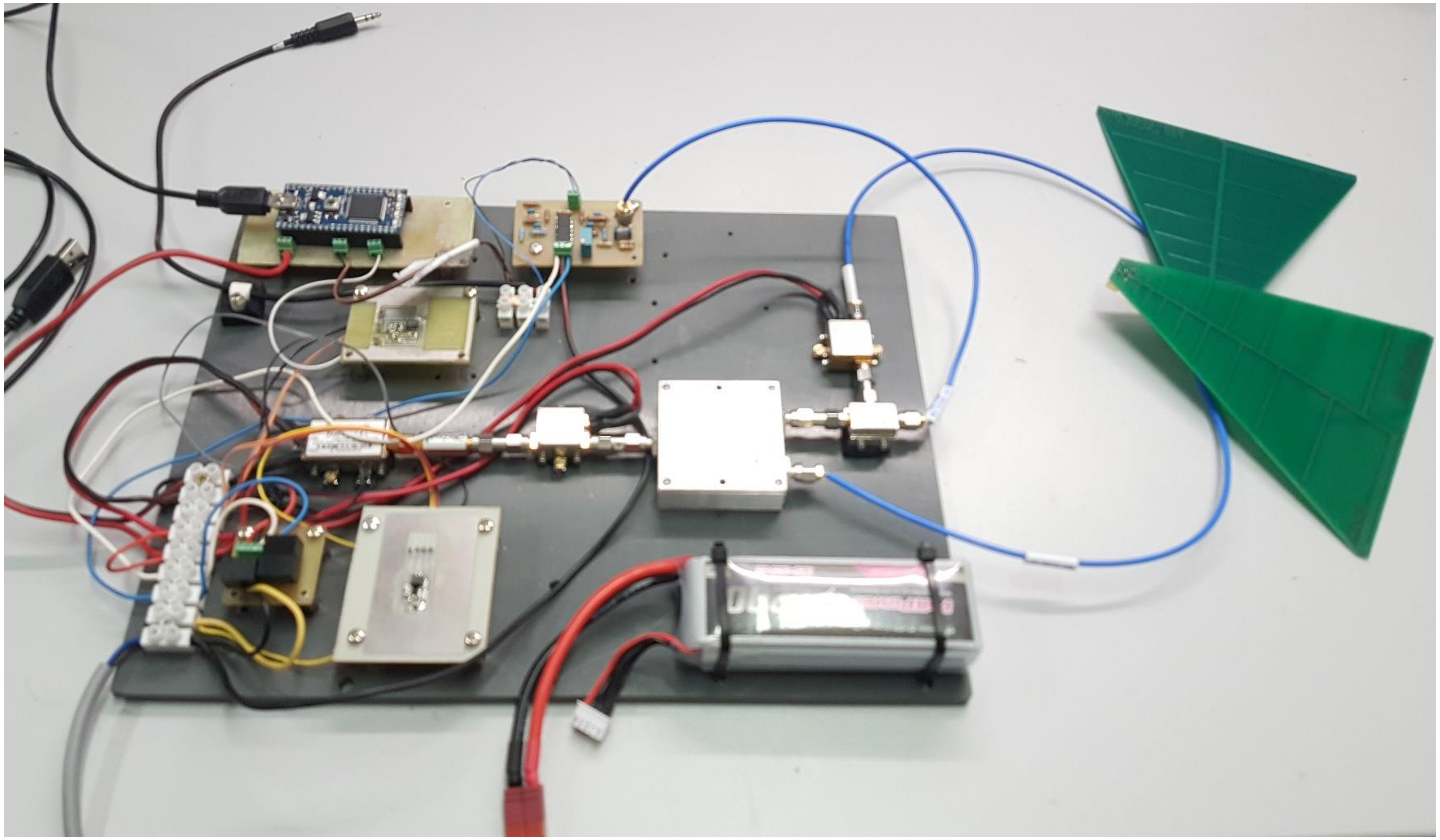
Guide

First assembly phases



Guide

The assembled hardware



COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware

to manage microcontroller

COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware to manage microcontroller

Developed software codes:

- Firmware program providing control functionalities of the Mbed NXP LPC1768 prototyping board. It provide the functionalities needed for the proper functioning of the GPR, as generating or changing selected waveforms. Moreover, it allow the communications between the user's host PC and the peripheral radar system, permitting an interactive control of GPR and data exchange.

- Software program for the FMCW GPR prototype in the form of graphical user interface (GUI) allowing simpler and more convenient user interaction with the system. The graphical interface provides an easy to use workspace, the user can switch many parameters to control the radar system and send them to the microcontroller.

Guide

Software and Firmware to manage microcontroller

Software development kit (SDK), operating system (OS), and library:

- For the firmware: it was written with the open source embedded operating system Arm Mbed OS [1], designed specifically for the "things" in the Internet of Things based on an Arm Cortex-M microcontroller. It provides C/C++ software development kit (SDK) with libraries to build various applications. This OS includes many features such as: security, connectivity, an RTOS (real time OS), and drivers for sensors and I/O devices.

- For the graphical user interface (GUI): Python programming language was chosen. Tkinter[2] package was selected as Python library for the creation of the GUI. This toolkit was chosen for its simplicity, providing detailed tutorials and ready layouts. Furthermore, a Python development environment is needed to develop the software as Spider 3.1 provided by Anaconda, popular Python distribution

COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware to manage microcontroller

How to find and get SDK, OS, package and library:

- Arm Mbed OS, which includes Cortex Microcontroller Software Interface Standard (CMSIS) package, and provides C/C++ SDK with proper libraries, can be used by accessing to the web site
https://os.mbed.com/
and selecting the section "Compiler", after the free account login.

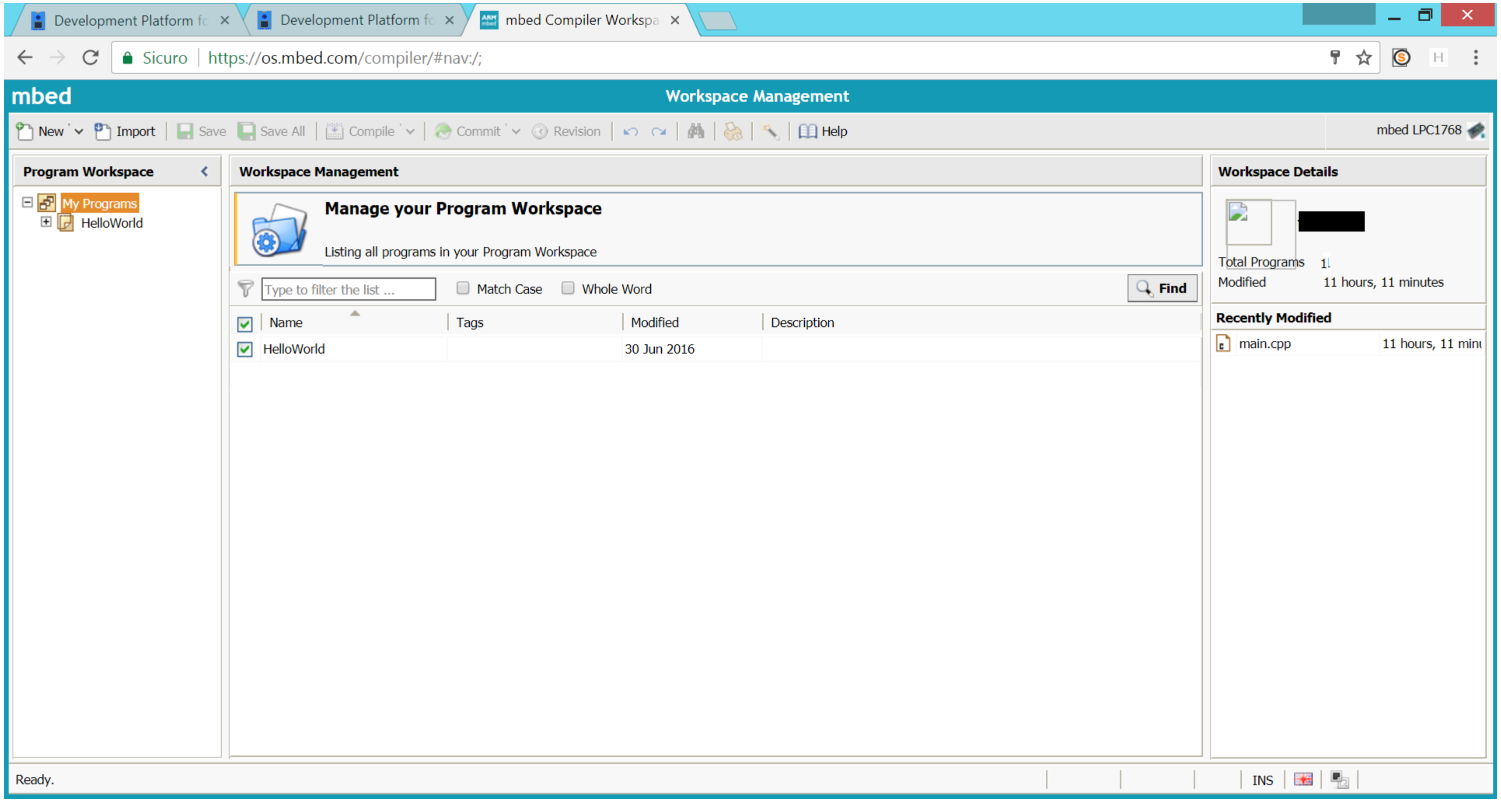
- Anaconda distribution, can be download from the web site:
https://www.anaconda.com/download/
Anaconda provides Spider workspace but is still required to get the Python 3.6 documentation, that includes Tkinter, from the web site
https://www.python.org/downloads/

- MbedWinSerial is the driver for the serial data transmission between user PC port and Mbed board, it can be downloaded from the website:
https://os.mbed.com/handbook/Windows-serial-configuration#1-download-the-mbed-windows-serial-port

Guide

Software and Firmware to manage microcontroller

The Mbed SDK allows to write, compile, and build C/C++ codes.



[redacted]

Total Programs 1
Modified 11 hours, 11 minutes

main.cpp 11 hours, 11 minu

☑	Name	Tags	Modified	Description
☑	HelloWorld		30 Jun 2016	

The .bin file as a result of a compilation should be copied into the Mbed microcontroller, connected by USB.

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

Serial communication: USB (Universal Serial Bus) was selected for communication between the user's host PC and the peripheral radar system. To establish the serial connection Mbed, microprogrammed control unit (MCU) was configured as a USB Virtual Serial Port with default connection settings of 9600 baud, 8 bits, 1 stop bit, no parity data packing. LPC1768 Mbed-board serial driver for Windows was installed onto the workspace PC.

The instructions at the beginning of the code in the main file (Mbed_Python_Control_LPC1768_2_main.txt) call the mbed.h library and open the communication channel between the Mbed and the host PC through a USB Virtual Serial Port. The project was created in language C++ and have to be compiled in on-line compiler on the official Mbed website[1] because the mbed.h include the operating system Arm Mbed OS that need to be open with the dedicated software:

```
#include "mbed.h"
Serial pc(USBTX, USBRX);
```

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

Switch waveform :

To obtain a signal with a desired waveform: either triangular, rectangular or saw-tooth, which means that the VCO frequency should change in time with a defined period T in the selected range $[F_{min}/F_{max}]$ corresponding to the DAC tuning voltage of V_{min}/V_{max}.

From the VCO ZX95-2700A+ documentation the coefficient of no-linear dependency of the voltage on the output frequency was calculated. At the normal temperature of 25°C for V= 0V the output is F_{min}= 1216.7 MHz and for V=25V $F_{max} = 2902.5\text{MHz}$.

Considering that the voltage amplifier ZX60-P105LN+ has a maximum frequency range at 2600 MHz instead of 2700MHz of VCO, there is an intrinsic limitation on the maximum value of frequency reachable on the user chosen frequency. It is due to the characteristic of this broadband voltage amplifier, so the possible interval of real frequency will be between 1216.7 MHz and 2600 MHz achievable with a voltage range of 0V-17V.

COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

Switch waveform :

Using these parameters can be calculated the required control voltage for each selected frequency. The mbed's LPC1768 chip has a 10-bit DAC. For each digital value input to the DAC, there is a corresponding analog output value.

For each digital value input to the DAC, there is a corresponding analog output value given by $V = (\frac{D}{2^n})(V_{max}\text{-}V_{min})$,

where V_{min}, V_{max} are maximum and minimum output voltages, *D* is the digital input, for the 10-bit DAC, $2^{10} = 1024$. The step size, or resolution, is therefore 3.3/1024, i.e. 3.2mV per bit [7]

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

DAC calculation: Mbed OS provides functions for various pins and interfaces configurations. Thus, it provides *AnalogOut* function to utilize the embedded DAC. *AnalogOut* function, by default, takes as an analog object a floating point number between 0.0 and 1.0 and outputs it to the single LPC1768 analog output pin – pin 18. The actual output voltage on pin 18 is between 0V and 3.3V, so the floating point number that is output as voltage is scaled by a factor of 3.3.

On the PC-side in the Python language a function was created for calculating the tuning voltage values of DAC for the selected waveform, bounding frequencies , in range F 1216.7–2902.5 MHz and the signal period *T* = 10–100 ms.

The calculated tuning voltage values and the selected signal period value were packed into a string in ascii format and transferred to the Mbed MCU through the Serial connection with the help of another realized data packing and transferring function. On the Mbed-side a universal function was created to receive and parse data from the PC.

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

DAC Update: It was decided to use one of the 26 available on the Mbed board GPIO pins for generating a square wave with which the measured radar signal would be synchronized. For implementing the synchronizing signal *DigitalOut* function provided by Mbed OS was used, which sets the state of the output pin (pin 25 in this case) and reads back the current output state.

To form a regular signal the DAC values should be updated at regular time intervals, which is usually accomplished by using the MCU clock and interrupt routines. In this project, two functions for triggering the signal were tested, *wait* and *Ticker*, the latter is used to setup a recurring interrupt to repeatedly call a function at a specified rate. Testing these two functions at different period values of the signal with the help of a Tektronix oscilloscope showed that *Ticker* provides time resolution of tens of microseconds, while *wait* – only of a hundred microseconds.

COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

DAC Update: . For *Ticker* at 10 us period the signal is stable, the measured period standard deviation is less than 200 picosecond. Therefore, the obtained square wave on the GPIO has a high time resolution of 10 us, fully meeting the specified requirement of 100 us, and can be applied for synchronization in the radar system.

Main Firmware: The final firmware project is organized as a main cycle with interrupts. In the main cycle the program waits for data transmission initialized from the GUI on the user's PC, it parses the data package into the DAC buffer values and the signal period value according to the predefined format. After the parsing, the *Ticker* interrupt routine is called with the read parameters. During the period *T* every *T/N* microseconds *Ticker* updates the DAC value according to the DAC buffer, where *N* is the buffer size. After *T* microseconds, the cycle counter is restarted and a new period of the radar signal is transmitted.

COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

Main Firmware:

(Mbed_Python_Control_LPC1768_2_main.txt)

The project was designed in the way that the firmware program does not depend on the radar signal parameters chosen by the user, it follows the same instructions and receives data in universal format for each radar signal range and waveform defined in the user software.

Such design of the project encourages its further development, for example, a modification of the RF-chain part of the system will require making changes only in the software, implemented in Python, while keeping the firmware program unchanged.

Guide

Software and Firmware to manage microcontroller

Firmware implementation:

VCO Predistortion implementation:

A strong feature of the developed software program is the consideration of nonlinear dependence of the VCO output frequency on the tuning voltage. The Python program on the user side (Calc_DAC_with_Predistortion.py) provides functions for the DAC buffer calculation required for obtaining the desired waveform at the VCO output.

In these calculations, the created predistortion function was used, which maps the desired frequencies to the tuning voltage values performing linear interpolation of the actual coefficients specified in the VCO ZX95-2700A+ documentation. It should also be mentioned that DAC voltage values are scaled by a factor of 3.3/25.0 to consider the effect of the voltage amplifier switched between the DAC and VCO to enable generating the full range of VCO frequencies.

Guide

Software and Firmware to manage microcontroller

Software Implementation:

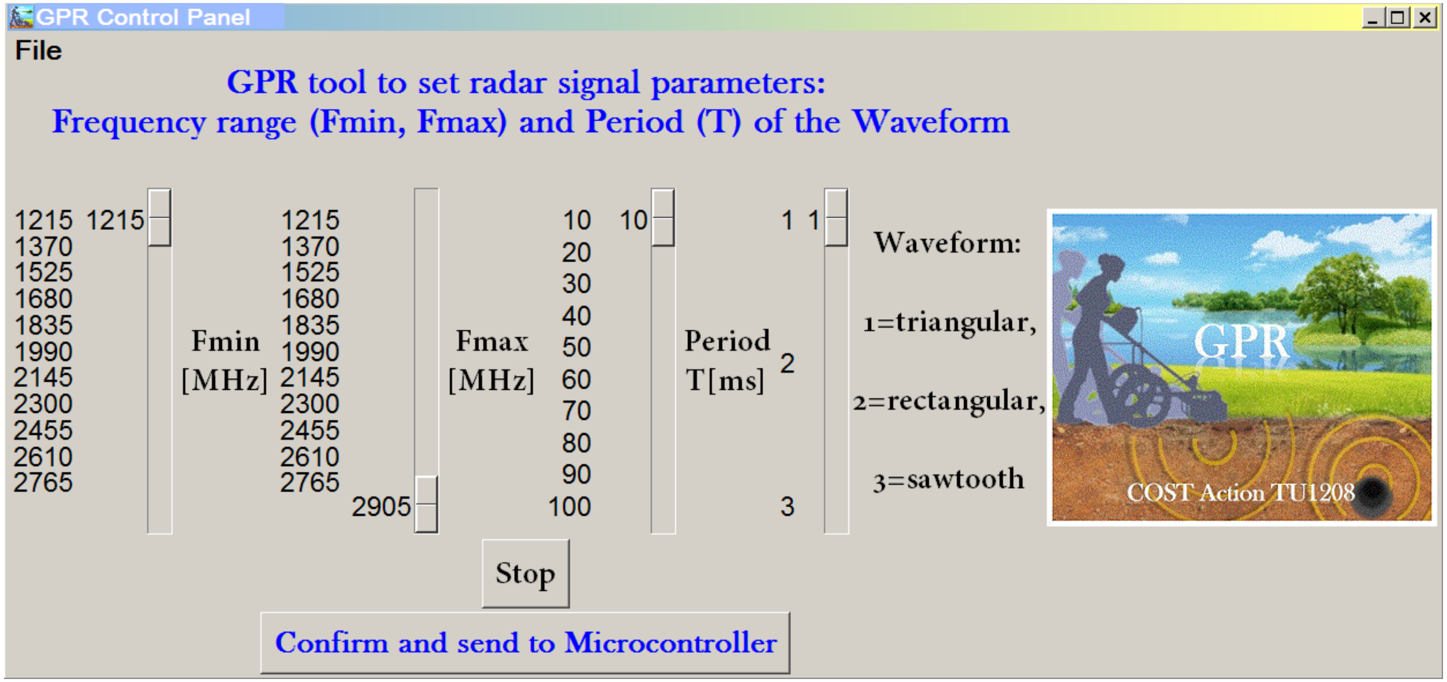
The developed GUI was aimed at providing interactive control of the radar system, specifically:

- choose the frequency range of the transmitted signal,
- select its period
- change the waveform
- arrest the transmission of the radar system

These first three control options were realized in the form of four sliders, moving the handlers of which the user can choose the desired parameters values.
The last control to stop the radar transmission was realized as a button to push in the low part of the window.
At the end of window there is a confirmation button, it allow to send the new control parameters to the Mbed
The main window of the designed GUI and the Help window containing instructions for the user.

Guide

Software and Firmware to manage microcontroller



GPR tool to set radar signal parameters:
Frequency range (Fmin, Fmax) and Period (T) of the Waveform

Fmin [MHz] 1215

Fmax [MHz] 2905

Period T[ms] 10

Waveform: 1

1=triangular,
2=rectangular,
3=sawtooth

Figure of the graphical interface dedicated to the FMCW GPR prototype control

Guide

Software and Firmware to manage microcontroller

Software Implementation:

In the developed GUI program, the change of a button’s state evokes a function which reads the sliders values, uses them in data buffer calculations, packs the data and sends them to the Mbed MCU.
The function also performs a check whether on the initial frequency of the signal set with the slider is lower than the final one, in case of violating this condition, a warning appears.
The Stop button at the GUI panel stops the radar signal transmission, in this case the program set the last fourth slider value as 4.
The developed software performs all the calculations on the user side, which increases the speed of radar data generation in comparison to the calculations on the side of Mbed MCU.
This program provides functions to compute the buffer of the DAC voltage values required to obtain the desired waveform at the VCO output and to pack the buffer into binary format and transmit it to Mbed MCU through the serial connection.
The file Python_GUI_4_Sliders.py contain the code of GUI program.

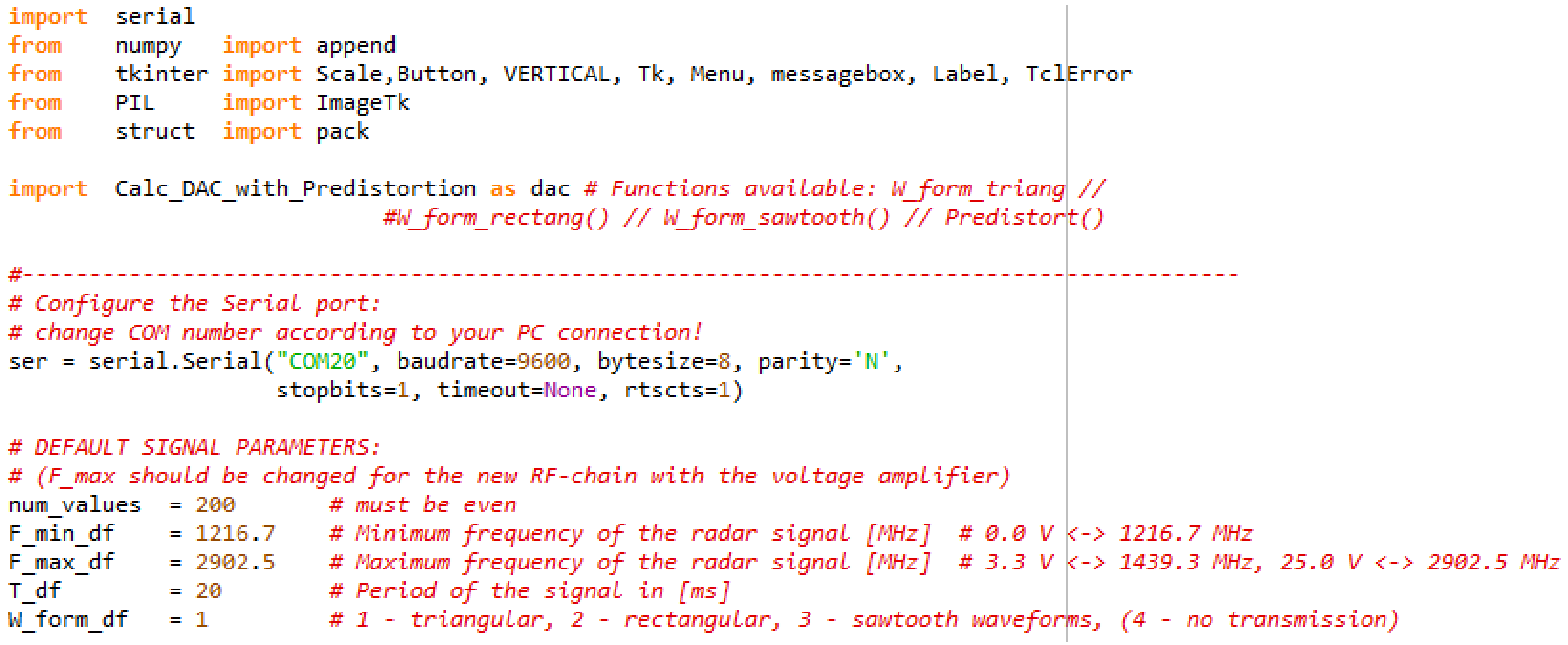
COST is supported by the
EU Framework Programme Horizon2020

Guide

Software and Firmware to manage microcontroller

The first part of the code is dedicated to import libraries and package as serial, that allow serial communications and the Tkinter package. Tkinter provide Python interface to the Tk GUI toolkit. Also, it have to be imported some *Tkinter* modules. After, the serial port have to be configured and the COM port have to be set.

Carrying on, the signal parameters value was inizialized to set as default value of the four slide. The fourth slide is set as value 1 for triangular wave transmission but it can be switched as value 2 (rectangular), 3 (sawthooth) or 4 to stop radar transmission.



```
import   serial
from     numpy   import append
from     tkinter import Scale,Button, VERTICAL, Tk, Menu, messagebox, Label, TclError
from     PIL     import ImageTk
from     struct  import pack

import   Calc_DAC_with_Predistortion as dac # Functions available: W_form_triang //
                            #W_form_rectang() // W_form_sawtooth() // Predistort()

#--------------------------------------------------------------------------------------------
# Configure the Serial port:
# change COM number according to your PC connection!
ser = serial.Serial("COM20", baudrate=9600, bytesize=8, parity='N',
                    stopbits=1, timeout=None, rtscts=1)

# DEFAULT SIGNAL PARAMETERS:
# (F_max should be changed for the new RF-chain with the voltage amplifier)
num_values  = 200       # must be even
F_min_df    = 1216.7    # Minimum frequency of the radar signal [MHz]  # 0.0 V <-> 1216.7 MHz
F_max_df    = 2902.5    # Maximum frequency of the radar signal [MHz]  # 3.3 V <-> 1439.3 MHz, 25.0 V <-> 2902.5 MHz
T_df        = 20        # Period of the signal in [ms]
W_form_df   = 1         # 1 - triangular, 2 - rectangular, 3 - sawtooth waveforms, (4 - no transmission)
```

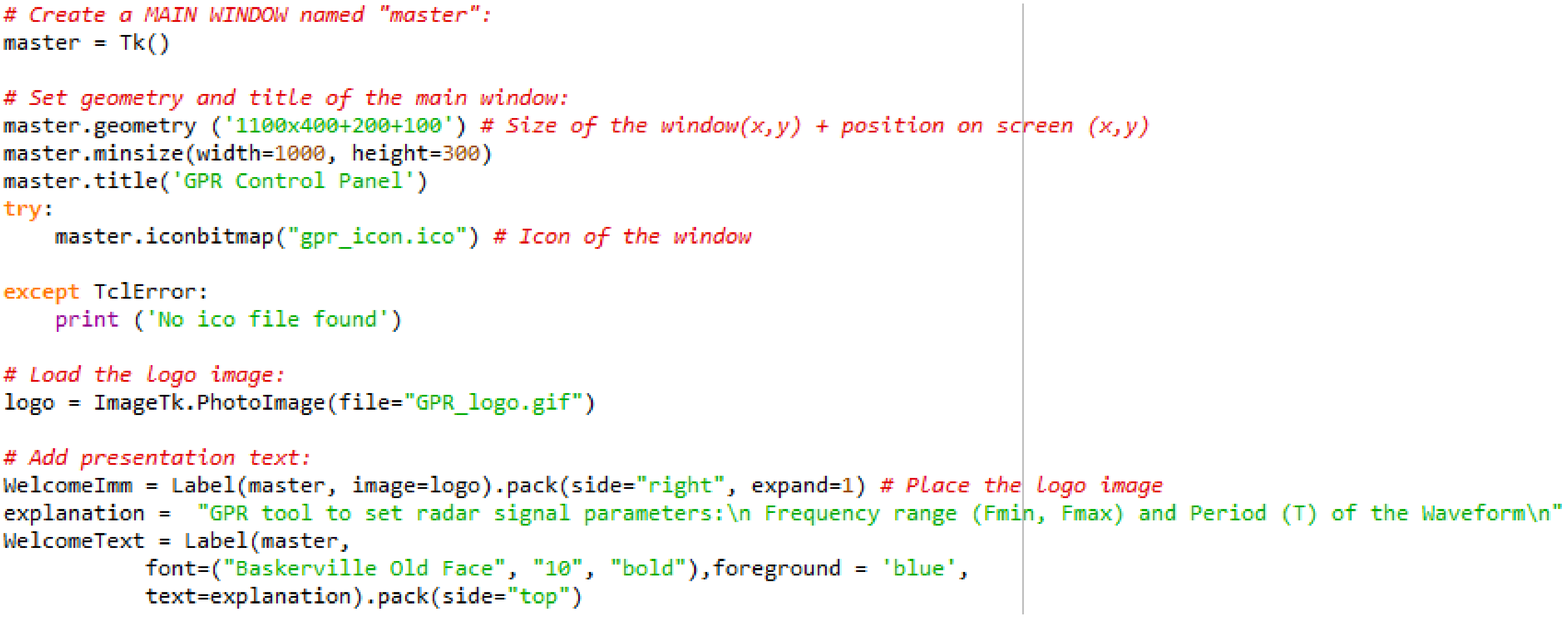
Guide

Software and Firmware to manage microcontroller

The Graphical interface is easy to create using Tkinter functionalities, starting to set the *main windows* size, name, presentation text and images.

Be sure to insert the selected icon with the proper name inside the same folder of the Phyton project, if not an exeption will start.

All the font, size, icons and image name is easy to change inside the code.

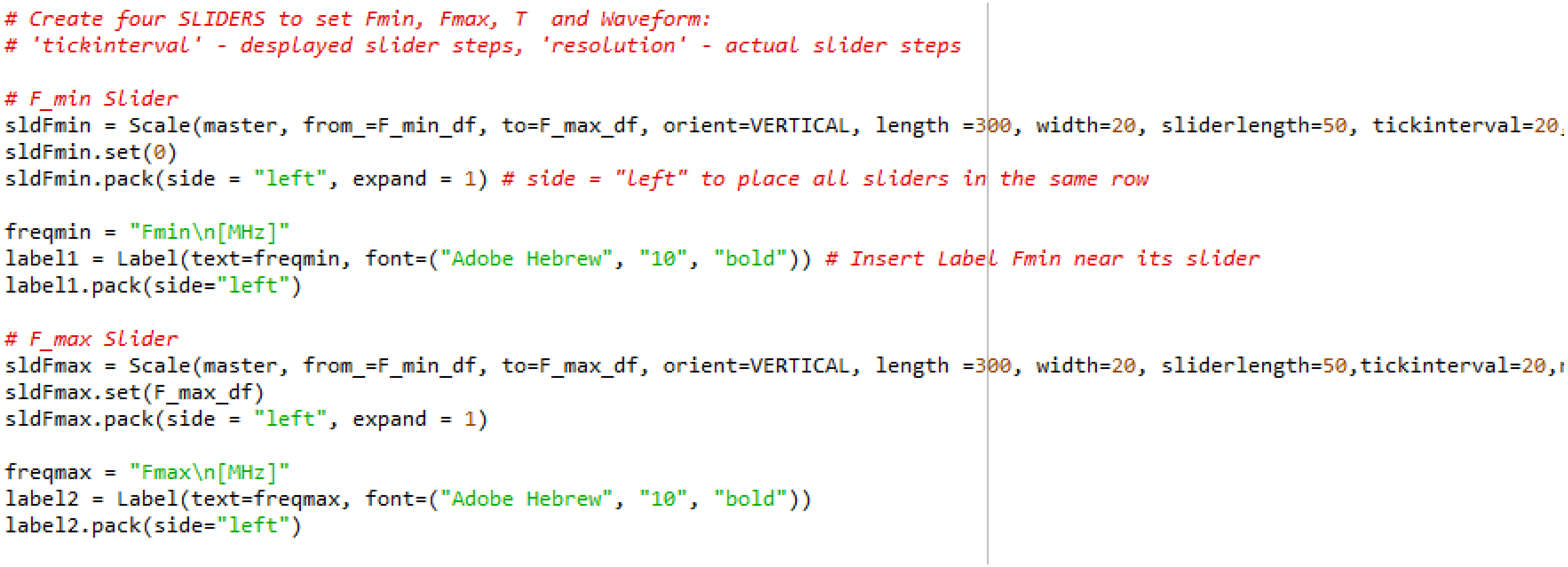


```
# Create a MAIN WINDOW named "master":
master = Tk()

# Set geometry and title of the main window:
master.geometry ('1100x400+200+100') # Size of the window(x,y) + position on screen (x,y)
master.minsize(width=1000, height=300)
master.title('GPR Control Panel')
try:
    master.iconbitmap("gpr_icon.ico") # Icon of the window

except TclError:
    print ('No ico file found')

# Load the logo image:
logo = ImageTk.PhotoImage(file="GPR_logo.gif")

# Add presentation text:
WelcomeImm = Label(master, image=logo).pack(side="right", expand=1) # Place the logo image
explanation =  "GPR tool to set radar signal parameters:\n Frequency range (Fmin, Fmax) and Period (T) of the Waveform\n"
WelcomeText = Label(master,
           font=("Baskerville Old Face", "10", "bold"),foreground = 'blue',
           text=explanation).pack(side="top")
```

Guide

Software and Firmware to manage microcontroller

Sliders creation: the creation of the graphical interface of the four sliders is possible setting various parameters needed as the orientation, the name, the position, the dimension and the tickinterval, the minimum value that can be changed as interval.
The instruction “from=” and “to=” is used to set the minimum and the maximum value of the variable.

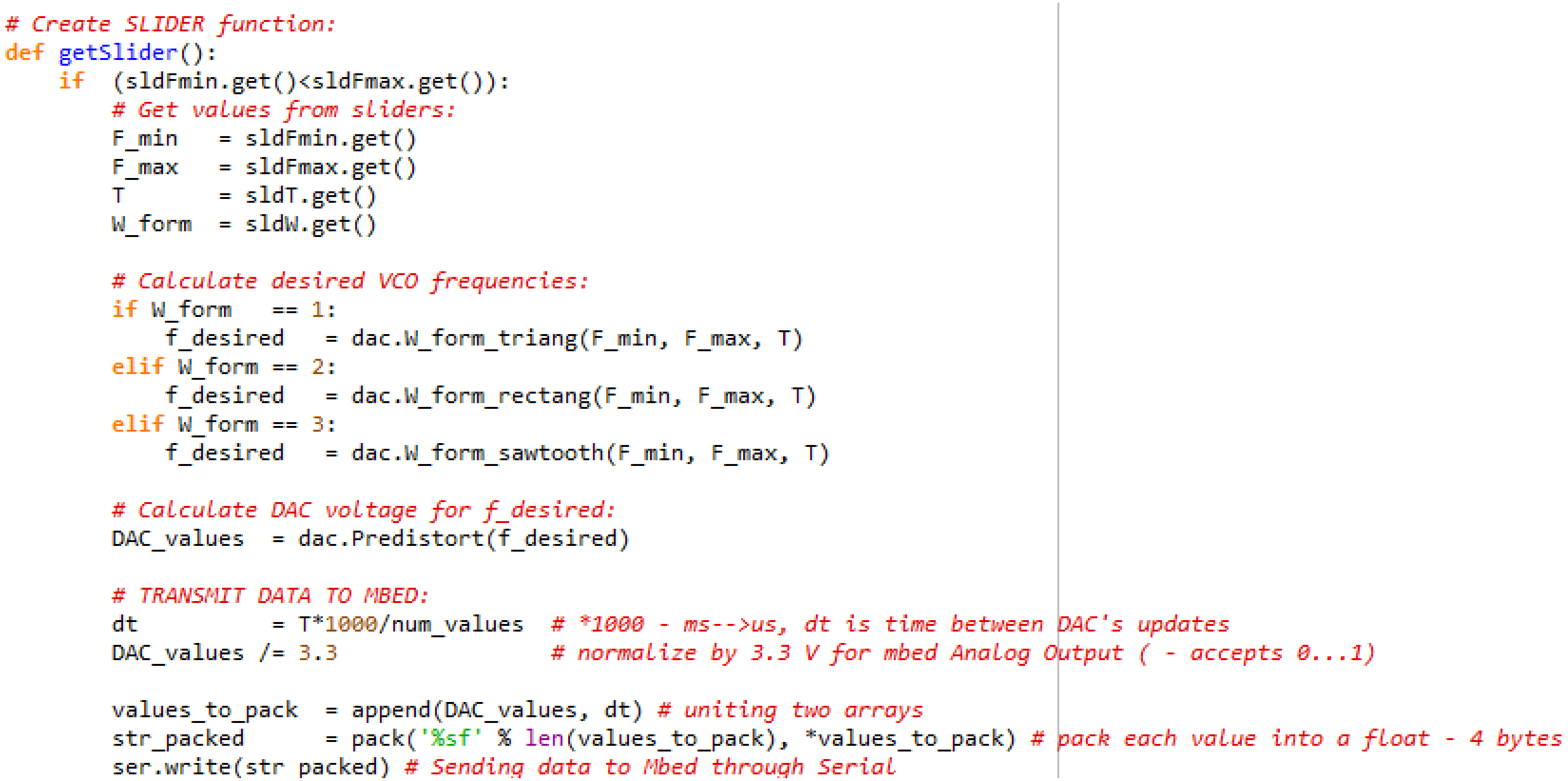


```
# Create four SLIDERS to set Fmin, Fmax, T  and Waveform:
# 'tickinterval' - desplayed slider steps, 'resolution' - actual slider steps

# F_min Slider
sldFmin = Scale(master, from_=F_min_df, to=F_max_df, orient=VERTICAL, length =300, width=20, sliderlength=50, tickinterval=20,
sldFmin.set(0)
sldFmin.pack(side = "left", expand = 1) # side = "left" to place all sliders in the same row

freqmin = "Fmin\n[MHz]"
label1 = Label(text=freqmin, font=("Adobe Hebrew", "10", "bold")) # Insert Label Fmin near its slider
label1.pack(side="left")

# F_max Slider
sldFmax = Scale(master, from_=F_min_df, to=F_max_df, orient=VERTICAL, length =300, width=20, sliderlength=50,tickinterval=20,
sldFmax.set(F_max_df)
sldFmax.pack(side = "left", expand = 1)

freqmax = "Fmax\n[MHz]"
label2 = Label(text=freqmax, font=("Adobe Hebrew", "10", "bold"))
label2.pack(side="left")
```

Guide

Software and Firmware to manage microcontroller

Sliders function: The getSlider function was built to gather the values set by the user. It provides functions to compute the buffer of the DAC voltage values required to obtain the desired waveform at the VCO output, to pack the buffer into binary format and transmit it to Mbed MCU through the serial connection.



```
# Create SLIDER function:
def getSlider():
    if  (sldFmin.get()<sldFmax.get()):
        # Get values from sliders:
        F_min   = sldFmin.get()
        F_max   = sldFmax.get()
        T       = sldT.get()
        W_form  = sldW.get()

        # Calculate desired VCO frequencies:
        if W_form   == 1:
            f_desired   = dac.W_form_triang(F_min, F_max, T)
        elif W_form == 2:
            f_desired   = dac.W_form_rectang(F_min, F_max, T)
        elif W_form == 3:
            f_desired   = dac.W_form_sawtooth(F_min, F_max, T)

        # Calculate DAC voltage for f_desired:
        DAC_values  = dac.Predistort(f_desired)

        # TRANSMIT DATA TO MBED:
        dt          = T*1000/num_values  # *1000 - ms-->us, dt is time between DAC's updates
        DAC_values /= 3.3                # normalize by 3.3 V for mbed Analog Output ( - accepts 0...1)

        values_to_pack  = append(DAC_values, dt) # uniting two arrays
        str_packed      = pack('%sf' % len(values_to_pack), *values_to_pack) # pack each value into a float - 4 bytes
        ser.write(str packed) # Sending data to Mbed through Serial
```

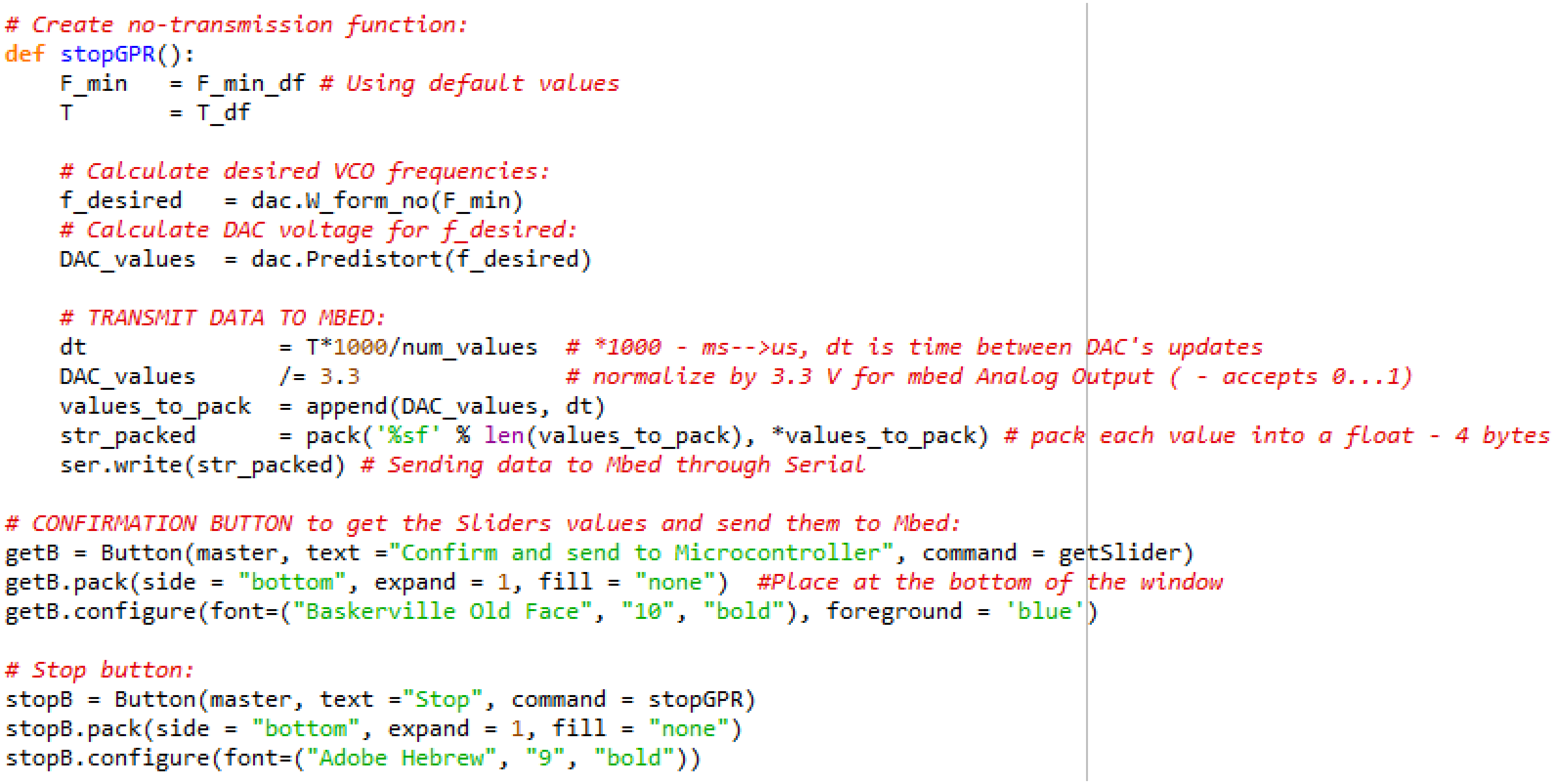
Guide

Software and Firmware to manage microcontroller

The creation of buttons is similar to the sliders creation and the confirmation button can send the new instruction to Mbed getting it with the command getSlider.

In the same way the command stopGPR of the Stop button recall the function stopGPR that can allow to block the radar transmission of waveform.

A the end of the entire code an instruction start the main loop: “master.mainloop()”



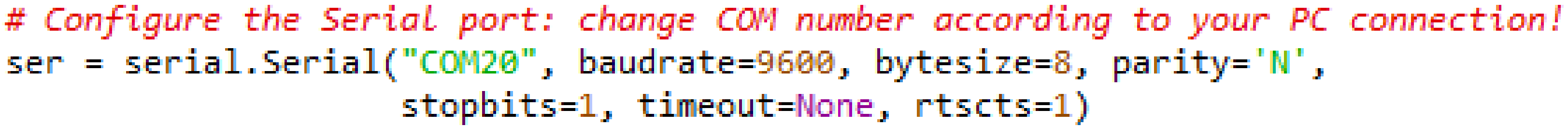
```
# Create no-transmission function:
def stopGPR():
    F_min   = F_min_df # Using default values
    T       = T_df

    # Calculate desired VCO frequencies:
    f_desired   = dac.W_form_no(F_min)
    # Calculate DAC voltage for f_desired:
    DAC_values  = dac.Predistort(f_desired)

    # TRANSMIT DATA TO MBED:
    dt              = T*1000/num_values  # *1000 - ms-->us, dt is time between DAC's updates
    DAC_values      /= 3.3               # normalize by 3.3 V for mbed Analog Output ( - accepts 0...1)
    values_to_pack  = append(DAC_values, dt)
    str_packed      = pack('%sf' % len(values_to_pack), *values_to_pack) # pack each value into a float - 4 bytes
    ser.write(str_packed) # Sending data to Mbed through Serial

# CONFIRMATION BUTTON to get the Sliders values and send them to Mbed:
getB = Button(master, text ="Confirm and send to Microcontroller", command = getSlider)
getB.pack(side = "bottom", expand = 1, fill = "none")  #Place at the bottom of the window
getB.configure(font=("Baskerville Old Face", "10", "bold"), foreground = 'blue')

# Stop button:
stopB = Button(master, text ="Stop", command = stopGPR)
stopB.pack(side = "bottom", expand = 1, fill = "none")
stopB.configure(font=("Adobe Hebrew", "9", "bold"))
```

Guide

Software and Firmware to manage microcontroller

How to run the software:

- Download and install Phyton 3.6 documentation which includes *Tkinter* package
- Download and install Anaconda distribution that include Spider development environment
- Copy all the Python files (Python_GUI_4_Sliders.py and Calc_DAC_with_Predistortion.py) and the images (GPR_logo.gif and gpr_icon.ico) provided in the same single folder. All files are included in the zipped GPR.exe auto-extractable file.
- Connect the Mbed board. Download and install MbedWinSerial driver for serial data transmission to the user pc and check the number of COM port used in the system control panel.
- Open your shell command line and write the command to install serial package for Python: pip install pyserial
- Open the file Python_GUI_4_Sliders.py with Spider workspace program, than check the value of COM port and if is not the same in use in the user pc from Mbed, change it with the correct value:



```
# Configure the Serial port: change COM number according to your PC connection!
ser = serial.Serial("COM20", baudrate=9600, bytesize=8, parity='N',
                           stopbits=1, timeout=None, rtscts=1)
```

Guide

Software and Firmware to manage microcontroller

How to run the software:

- Register a free account on the Mbed OS web site and run the open-source Mbed SDK. Connect the Mbed board, create a new project and open the file Mbed_Python_Control_LPC1768_2_main.txt
- Download the Mbed official library from the web site: https://os.mbed.com/users/mbed_official/code/mbed/
- Import it into the on-line compiler in the same project, than compile the main file
- The .bin file as a result of the compilation should be copy into the Mbed microcontroller, connected by USB
- Run the file Python_GUI_4_Sliders.py with Spider. It opens the Graphical user interface of the GPR control panel. Select the desired parameters and waveform type than press the confirmation button to forward the commands to the Mbed board that manage and control the GPR
- The interactive control of the GPR is posssible using the GPR control panel allowing to switch parameters and waveform on-line or to stop the radar transmission everytime

COST is supported by the
EU Framework Programme Horizon2020

Guide

Basic software

for interpreting output signal

COST is supported by the
EU Framework Programme Horizon2020

Guide

Basic software for interpreting output signal

Many software development environment allow to manage output GPR signal coming at right audio channel. Python, MathWorks are examples of these development environments that provide specific libraries for estimating range of a detected target.

At present, e.g. the 2017 release of MathWorks define the typical high level procedure shown in a next slide. Note, the first step (Dechirp) has been carried out already by hardware subsystems Mixer+(Amplifier+Filter), and consequently the software for interpreting right audio channel should be start from “find beat frequency”, the second point of the procedure.

Before find beat frequency, acquire audio signal by using function such as
myRecording = getaudiodata(recObj);

With
recObj = audiorecorder(Fs_audio,Nr_bit_audio,NR_can_audio,ID);
Fs_audio is sampling rate
Nr_can_audio=2 and ID=0

Guide

Basic software for interpreting output signal

For the case of saw tooth need evaluation of one beat frequency, whereas for triangular modulation either up-sweep and down-sweep beat frequencies should be estimated.

Once extracted the frequency information, e.g. by means of *pwelch()* function, the *range2beat()* function allows the estimation of the range.

Doppler effect, and range-Doppler coupling can be evaluated using functions as

Phased.RangeDopplerResponse
rdcoupling(fd,slope,c)

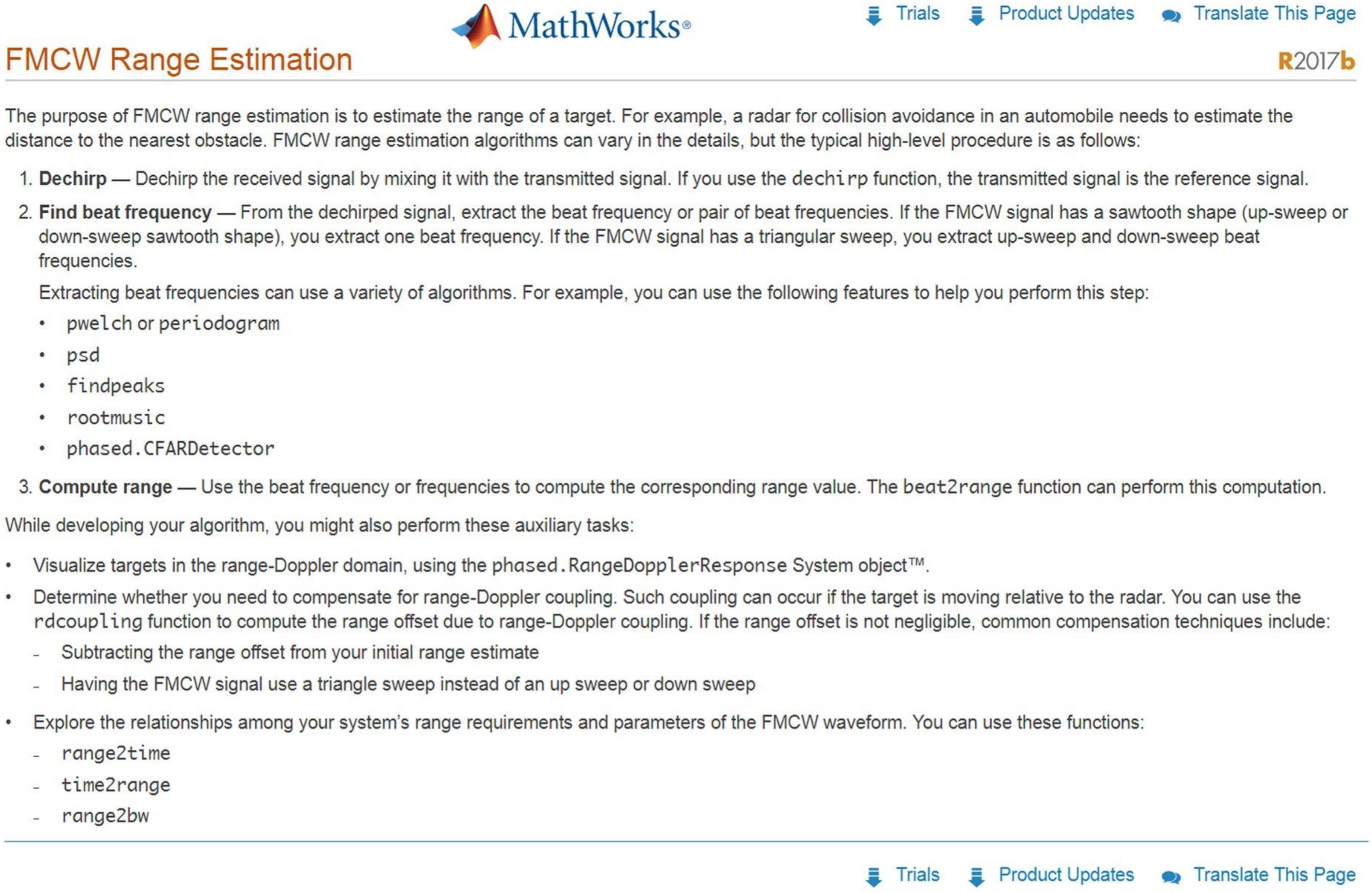
And so, calculate the range offset due to range-Doppler coupling.

Moreover, other functions help to find different parameters correlated with the signal, such as *range2bw*, *range2time*,….

COST is supported by the
EU Framework Programme Horizon2020

Guide

Basic software for interpreting output signal



FMCW Range Estimation

The purpose of FMCW range estimation is to estimate the range of a target. For example, a radar for collision avoidance in an automobile needs to estimate the distance to the nearest obstacle. FMCW range estimation algorithms can vary in the details, but the typical high-level procedure is as follows:

1. **Dechirp** — Dechirp the received signal by mixing it with the transmitted signal. If you use the `dechirp` function, the transmitted signal is the reference signal.
2. **Find beat frequency** — From the dechirped signal, extract the beat frequency or pair of beat frequencies. If the FMCW signal has a sawtooth shape (up-sweep or down-sweep sawtooth shape), you extract one beat frequency. If the FMCW signal has a triangular sweep, you extract up-sweep and down-sweep beat frequencies.

   Extracting beat frequencies can use a variety of algorithms. For example, you can use the following features to help you perform this step:

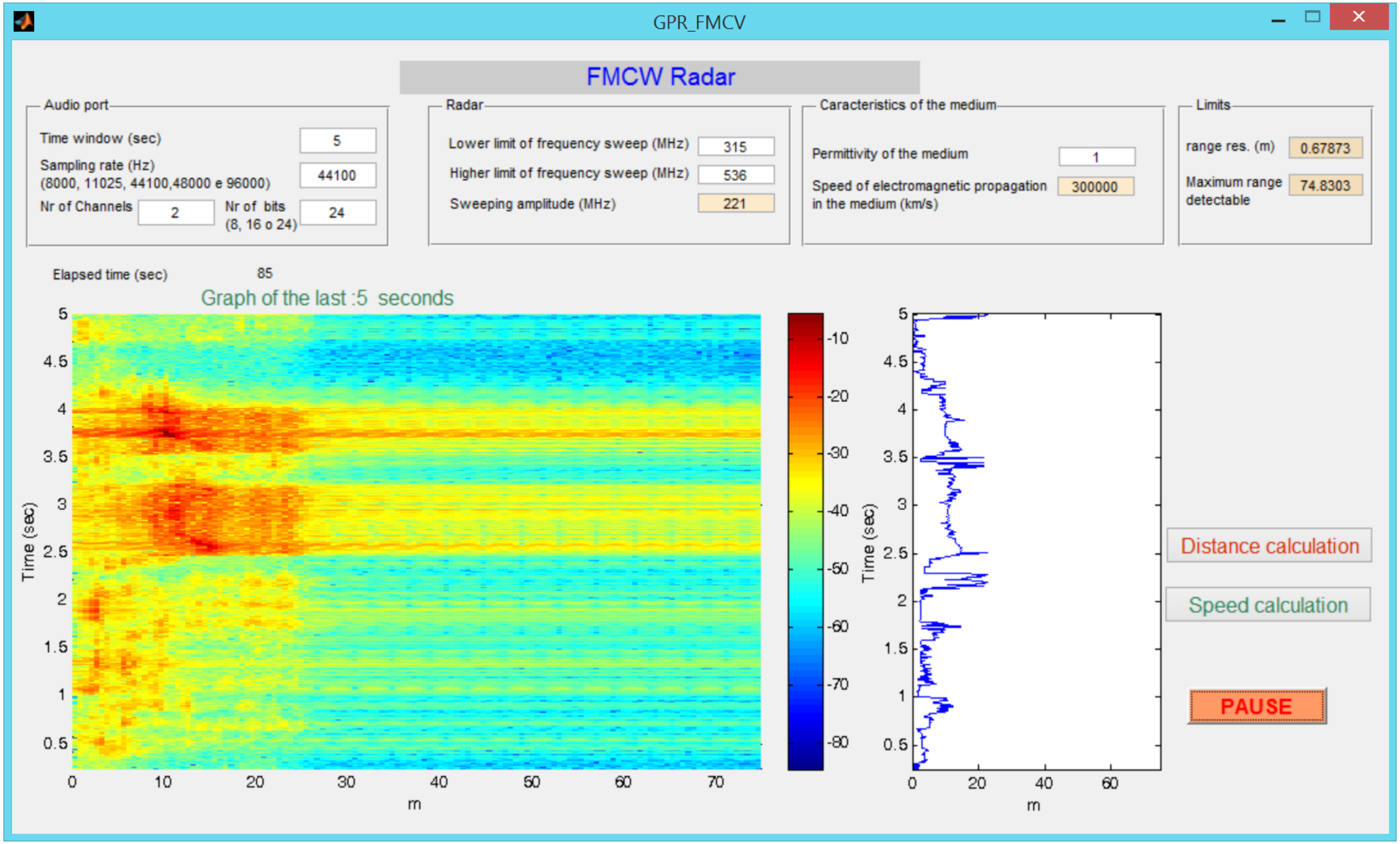
   - `pwelch` or `periodogram`
   - `psd`
   - `findpeaks`
   - `rootmusic`
   - `phased.CFARDetector`
3. **Compute range** — Use the beat frequency or frequencies to compute the corresponding range value. The `beat2range` function can perform this computation.

While developing your algorithm, you might also perform these auxiliary tasks:

- Visualize targets in the range-Doppler domain, using the `phased.RangeDopplerResponse` System object™.
- Determine whether you need to compensate for range-Doppler coupling. Such coupling can occur if the target is moving relative to the radar. You can use the `rdcoupling` function to compute the range offset due to range-Doppler coupling. If the range offset is not negligible, common compensation techniques include:
  - Subtracting the range offset from your initial range estimate
  - Having the FMCW signal use a triangle sweep instead of an up sweep or down sweep
- Explore the relationships among your system's range requirements and parameters of the FMCW waveform. You can use these functions:
  - `range2time`
  - `time2range`
  - `range2bw`

Guide

Basic software for interpreting output signal



GPR_FMCV

FMCW Radar

Audio port

Time window (sec) 5

Sampling rate (Hz) (8000, 11025, 44100,48000 e 96000) 44100

Nr of Channels 2

Nr of bits (8, 16 o 24) 24

Radar

Lower limit of frequency sweep (MHz) 315

Higher limit of frequency sweep (MHz) 536

Sweeping amplitude (MHz) 221

Caracteristics of the medium

Permittivity of the medium 1

Speed of electromagnetic propagation in the medium (km/s) 300000

Limits

range res. (m) 0.67873

Maximum range detectable 74.8303

Elapsed time (sec) 85

Graph of the last :5 seconds

Recommendations

Recommendations

Hardware

- In order to simplify realization of PCBs, we used single layer PCBs. Normally each electronic board include component of the same mounting typology (as SMD). In the case of SMD components, mount them directly on the top layer. In the case of DIP components, holes for inserting the component from the opposite side of the copper layer must be made, so as to make welding to the ends of pins.

Conclusions

COST is supported by the
EU Framework Programme Horizon2020

Conclusions

A full guide to project and develop an open-source low-cost GPR radar for educational purpose is available on-line.

Hardware, firmware and software project and implementation guideline are available with full explanation of designs, practical operational and comments on the entire code.

The developed software and firmware programs allow interactive control of the GPR functioning, the implemented GUI program is user-friendly and will facilitate students' interaction with the radar system.

The total price of the radar does not exceed **600 euros**, however it can be further reduced with the use of more cheaper smd RF component, but in this case using more complex multilayer pcb. Anyway, the actual version of the radar is the more easy-to-learn and easy-to-built possible.

Moreover, reducing the bandwidth of frequency sweep, the price can be cut until 400 euros.

Future work will include designing dedicated antennas and proper radar kit form-factor.

References

[1] Mbed OS: https://www.mbed.com/en/development/mbed-os/

[2] Tkinter package: https://docs.python.org/2/library/tkinter.html

[3] V. Ferrara, F. Frezza, F. Mangini, E. Piuzzi, P. Simeoni, N. Tedeschi, *Design and realization of a cheap GPR prototype & Recent advancements in forward--scattering methods*, COST Action TU1208, 4th General Meeting, Athens, Greece, October19-20, 2015

[4] V. Ferrara, F. Troiani, F. Frezza, F. Mangini, L. Pajewski, P. Simeoni, N.Tedeschi, *Design and Realization of a Cheap Ground Penetrating Radar Prototype @ 2.45 GHz," IEEE Browse Conference Publications - 2016 10th European Conference on Antennas and Propagation (EuCAP), Davos, Switzerland, 10-15 April 2016, pp. 1-4, DOI: 10.1109/EuCAP.2016.7482008*

[5] La. Pajewski, R. Persico, V. Ferrara, S. Chicarella, F. Frezza, F. Troiani, "Ground Penetrating Radar prototypes developed in COST Action TU1208, " In the proceedings of 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2016), 2016, Split, Croatia September 22-24, 2016

References

[6] Gregory Charvat, Jonathan Williams, Alan Fenn, Steve Kogon, and Jeffrey Herd. RES.LL-003 Build a Small Radar System Capable of Sensing Range, Doppler, and Synthetic Aperture Radar Imaging. January IAP 2011. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

[7] Mbed DAC course-notes:
https://developer.mbed.org/media/uploads/robt/mbed_course_notes_-_analog_input_and_output.pdf

Additional Material

List of PCB layout files

COST is supported by the
EU Framework Programme Horizon2020

Additional Material

List of PCB layout files

Name	Description
Amp1_1_1.png	Low frequency amplifier (Amp1) subsystem
PCB videoampl_1_1.png	Amplifier+ Low pass filter (Amp2) subsystem
mbed_1_1.png	Mbed board
DC_1_PCB_1_1.png	Power supply subsystem (DC-DC converters). PCB design of circuit 1 (output +5V,±6.5V, GND)
DC_2_PCB_1_1.png	Power supply subsystem (DC-DC converters). PCB design of circuit 2 (output +27 V)

COST is supported by the
EU Framework Programme Horizon2020

Additional Material

List of datasheets

COST is supported by the
EU Framework Programme Horizon2020

Additional Material

List of datasheets

Name	Description	Subsystem
ZX95-2700A+.pdf	Voltage Controlled Oscillator Wide Band 1300 to 2700 MHz	RF
VAT-3+.pdf	SMA Fixed Attenuator 50Ω 1W 3dB DC to 6000 MHz	RF
ZX60-P105LN+.pdf	Low Noise Amplifier 50Ω 40 to 2600 MHz	RF
ZAPD-2-272+.pdf	Power Splitter/Combiner 2 Way-0° 50Ω 800 to 2700 MHz	RF
ZX05-43MH+.pdf	Frequency Mixer wide band Level 13 (LO Power +13 dBm) 824 to 4200 MHz	RF
SM-SM50+.pdf	Adapter, SMA-M to SMA-M 50Ω DC to 18 GHz	RF
086-12SM+.pdf	Coaxial Cable 50Ω DC to 18 GHz	RF
LT1077fa.pdf	Micropower, Single Supply, Precision Op Amp	Low frequency amplifier subsystem
OP467.pdf	Quad Precision, High Speed Operational Amplifier IC OPAMP GP 28MHZ 14DIP (OP467GP)	Amplifier+ Low pass filter (Amp2) subsystem

COST is supported by the
EU Framework Programme Horizon2020

Additional Material

List of datasheets - continue

Name	Description	Subsystem
LPC1768.pdf	Rapid prototyping for the LPC1768 MCU	Mbed board
mbed-005.1.pdf	List of the pins of mbed microcontroller	Mbed board
lpc17xx_um.pdf	LPC1768 user manual	Mbed board
lpc17xx_ds.pdf	LPC1768 datasheet	Mbed board
pc17xx_es.pdf	LPC1768 errata sheet	Mbed board
TSRN-1 DC_DC converter.pdf	DC/DC converter 6.5V	Power supply subsystem (DC-DC converters)
R-78Exx-0.5 DC_DC converter.pdf	Conv DC/DC 5V 500MA OUT THRU	Power supply subsystem (DC-DC converters)
LT3461Afa.pdf	(Step Up) DC-DC, Reg. 2.5V-16Vin, 1.25V-38Vout, 250mA out, TSOT-23-5	Power supply subsystem (DC-DC converters)

COST is supported by the
EU Framework Programme Horizon2020

Additional Material

Software and firmware codes list

COST is supported by the
EU Framework Programme Horizon2020

Additional Material

Software and firmware codes list

Name	Description
Python_GUI_4_Sliders.py	GPR control panel graphical interface
Calc_DAC_with_Predistortion.py	Imported in Python_GUI_4_Sliders.py
gpr_icon.ico	An icon of GPR, can be changed
GPR_logo.gif	An image for the control panel, can be changed
Mbed_Python_Control_LPC1768_2_main.txt	File that have to be upload on Mbed OS web site and than to be compiled to get a .bin file
Mbed_Python_Control_LPC1768_2.bin	Compiled .bin file

COST is supported by the
EU Framework Programme Horizon2020

Authors

Vincenzo Ferrara received the “Laurea” degree cum laude in Electronic Engineering in March 1980 from “La Sapienza” University of Rome. He joined the Department DIET of the same University, as Associate Professor of Electronic since 2001, and he is scientific responsible of Electronics for the Environment Lab of the DIET.

The scientific research conducted in the years after graduation concerns different topics: electronic systems for the environment, systems for the planning, design and management of broadcasting, geo-referenced detection and software models, application of new technologies for planning and sustainable development, satellite navigation, WSN low power low voltage, energy harvesting, design of RF circuits, LCD, porous silicon. Co-leader in the COST Action TU1208 Project 4.2 of first year of COST action: Advanced application of GPR to the localization and vital signs detection of buried and trapped people. Member of international scientific advisory committee of many conferences, as: Risk analysis, Sustainable Planning & Development, Brownfield, Disaster Management. TCP member of 2nd International workshop on collaborations in ERDM, 2014. (For more info, please visit https://sites.google.com/a/uniroma1.it/vincenzoferrara-eng/home).

Authors

Margarita Chizh is a PhD student at Bauman Moscow State Technical University. Her primary area of research is microwave imaging, focusing on radar signal processing and radar systems development. She is also a junior researcher at the Remote sensing laboratory which develops and produces holographic subsurface radars for various applications (For more info, please visit http://www.rslab.ru/english/).

Authors

Andrea Pietrelli received the B.S. degree in Automation Engineering from Sapienza, University of Rome, and the M.S. degree in Communications Engineering from Sapienza, University of Rome. Actually is a Ph.D. candidate on Electronics Engineering in joint supervision between D.I.E.T. department of Sapienza University of Rome and Laboratoire Ampere of École centrale de Lyon. His research areas include low power electronics, energy harvesting, microbial fuel cell, bioreactors, ground penetrating radar, wireless sensor network and radar system development.
(For more info, please visit
https://web.uniroma1.it/dip_diet/)