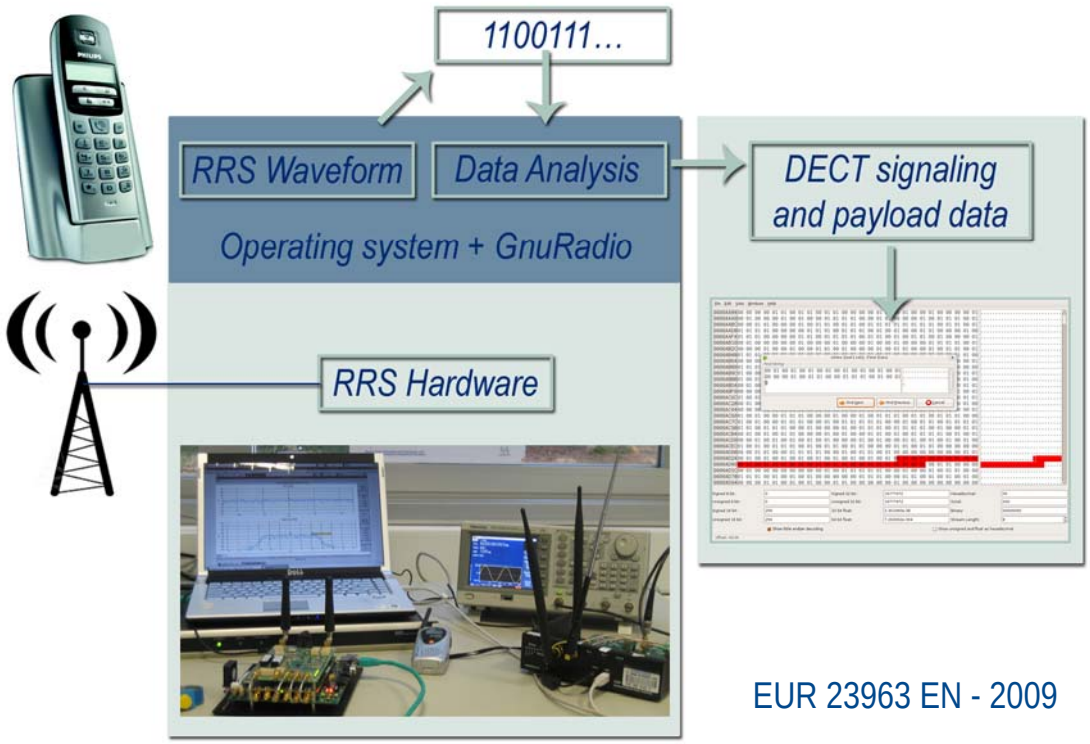
JRC Scientific and Technical Reports

# Reconfigurable Radio System
# Test bed for security research

**Raimondo Giuliani**
**Gianmarco Baldini**
**Dimitrios Symeonidis**




EUR 23963 EN - 2009

JRC EUROPEAN COMMISSION

ipsc Institute for the Protection and Security of the Citizen

The mission of the IPSC is to provide research results and to support EU policy-makers in their effort towards global security and towards protection of European citizens from accidents, deliberate attacks, fraud and illegal actions against EU policies.

European Commission
Joint Research Centre
Institute for the Protection and Security of the Citizen

**Contact information**
Address: Via Enrico Fermi 2749, TP723, 21027, Ispra, Varese
E-mail: gianmarco.baldini@jrc.ec.europa.eu
Tel.: +39 0332 78 6618
Fax: +39 0332 78 5469

http://ipsc.jrc.ec.europa.eu/
http://www.jrc.ec.europa.eu/



JRC 52750

EUR 23963 EN
ISSN 1018-5593

Luxembourg: Office for Official Publications of the European Communities



*Printed in Italy*

# TABLE of CONTENTS

# 1. Acronyms

| Acronym | Defined as |
|---|---|
| [ARI] | Access Rights Identity |
| [BER] | Bit Error Rate |
| [BS] | Base Station |
| [EU] | European Union |
| [ETSI] | European Telecommunications Standards Institute |
| [EVM] | Error Vector Magnitude |
| [FCC] | Federal Communications Commission |
| [HDAV] | High-Definition Audio and Video |
| [ICT] | Information and Communications Technologies |
| [IEEE] | Institute of Electrical and Electronics Engineers |
| [IP] | Internet Protocol |
| [ISO] | International Organization for Standardization |
| [ITU] | International Telecommunication Union |
| [LOS] | Line Of Sight |
| [OFDM] | Orthogonal Frequency Division Multiplexing |
| [NLOS] | Non-Line Of Sight |
| [OTA] | Over The Air |
| [PDU] | Protocol Data Unit |
| [PER] | Packet Error Rate |
| [PHY] | PHYsical (layer) |
| [RRS] | Reconfigurable Radio Systems |
| [SDO] | Standards Development Organizations |
| [SDR] | Software Defined Radio |
| [TPC] | Transmit Power Control |
| [TRP] | Total Radiated Power |
| [USA] | United States of America |
| [USB] | Universal Serial Bus |

# 2. Glossary

The following terms are used in the document:

*FP::Fixed Part (DECT Fixed Part) (FP)*: physical grouping that contains all of the elements in the DECT network between the local network and the DECT air interface (a DECT base station in plain English)

*RFP::Radio Fixed Part (RFP)*: one physical sub-group of a FP that contains all the radio end points (one or more) that are connected to a single system of antennas (a DECT radio in plain English)

*PP::Portable Part (DECT Portable Part) (PP)*: physical grouping that contains all elements between the user and the DECT air interface (a DECT phone in plain English)

*S field::Dect* packet preamble or synchronization field

*A field::Dect* packet signaling part

*B field::Dect* packet payload part

NOTE: the acronyms are from ETSI web site at http://www.etsi.org.

# 3. Introduction

Technological progress on the digital processing has opened the way to a novel implementation approach for wireless communication platforms where most of the digital signal processing is done in software rather than in hardware. Such systems have been known as Software Defined Radio (SDR) or Reconfigurable Radio Systems (RRS).

A typical SDR/RRS is able to execute all the radio frequency and base-band processing though software components rather then hardware components as in conventional radio communication systems. This capability provides a high level of reconfigurability and the possibility to implement a number of different algorithms for digital processing.
Therefore, SDR/RRS can be used for a variety or purposes including the possibility of implementing wireless security attacks against conventional communication systems.

In this technical report, we present an application of the SDR/RRS platform to implement a security attack against a DECT platform. The SDR/RRS platform has been used to implement a DECT demodulator and a processing module to eavesdrop and capture user and control data transmitted by a DECT system. The commercially available Universal Software Radio Peripheral (USRP) has been used as SDR/RRS platform for the development of the prototype.
The paper presents the technical challenges and implementation details in the development of the prototype and an overview of the capabilities of the USRP to implement wireless security attacks. The SDR/RRS platform used in the project is quite versatile and it can be used for a number of other applications related to DECT or other wireless communication systems.

The technical report is structured in the following parts:

1) The description of the concept of Software Defined Radio and Reconfigurable Radio Systems.
2) The description of the DECT standard.
3) Status of research and reference papers on the use of SDR/RRS to implement wireless security attacks.
4) Description of the SDR/RRS platform (USPR) used in the research and prototyping activity.
5) Description of the implementation of the prototype.
6) Description of the prototype execution and validation against a DECT system.
7) Future Developments.

# 4. Software Defined Radio and Reconfigurable Radio Systems

Reference [1] describes software-defined radio as:
"A software-defined radio (SDR) is one that has the capability – through use of programmable hardware (handsets) controlled by software - to tune to any frequency band and receive any modulation across a large frequency spectrum".

In the "Ideal Software Radio" the Digitalization starts right after the antenna (using D/A and A/D Converter with high dynamic) and the elaboration is implemented by DSP/FPGAs with very high throughput using different software for different waveforms.

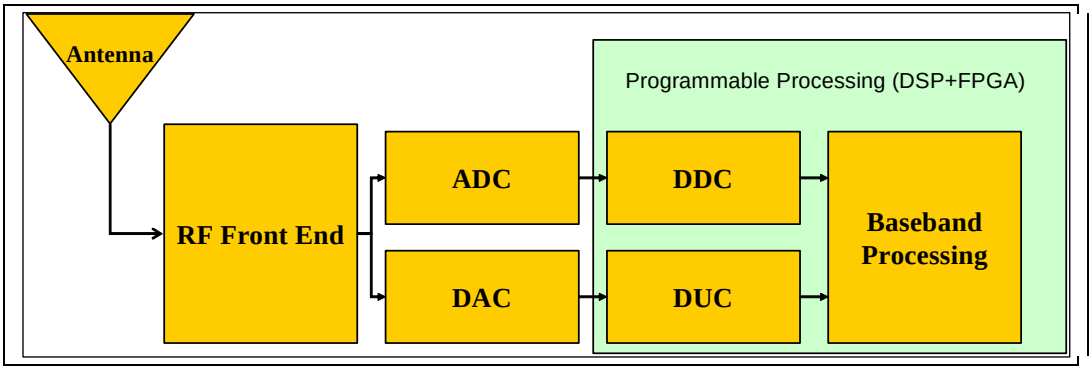



**Figure 1 Diagram of a Software Radio**

Theoretically, the hardware is able to identify the software with which it is being asked to interface and then to perform multiple tasks at the same time – in a similar way to a mobile telephone being able to act as a Global Positioning System (GPS), telephone and text sender simultaneously.
In other words, a SDR can receive and transmit a new form of radio protocol just by running new software. In this way, a SDR can reconfigure itself appropriately for its environment and can be quickly and easily upgraded over-the-air. SDRs can talk and listen to multiple channels at the same. The obvious advantage here is that the system can be changed by bringing in new application software without any change, replacement, or modification of handsets.
While in a radio conventional system, many functions are implemented in the hardware components, in a software defined radio, these functionalities are implemented in software as in the following figure:

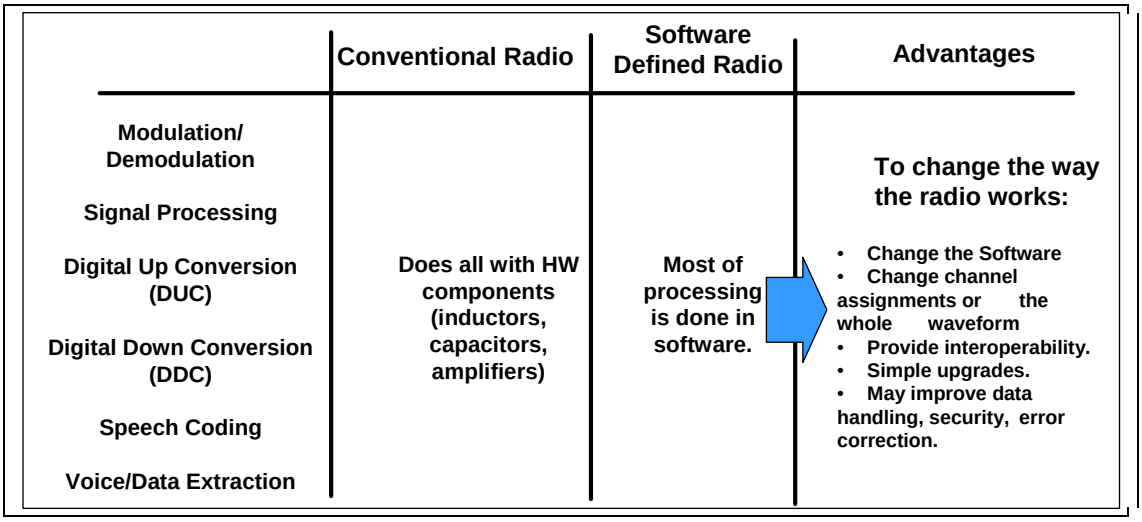



**Figure 2 Radio functions in a conventional and software defined radio.**

Software Defined Radio is a technology enabler for the Cognitive Radio, which is defined as a radio or wireless communication that is able to change dynamically its transmission or reception parameters by using the information collected or sensed on the external environment.

The term Software Defined Radio has been traditionally used to represent the SDR implemented in the Joint Tactical Radio Systems program by the US military. The JTRS (Joint Tactical Radio Systems) is intended to permit the military services to operate together in a "seamless" manner via wireless voice, video, and data communications through all levels of command, including direct access to near real-time information from airborne and battlefield sensors.
For a description of the JTRS program, please refer to [2].

JTRS has been traditionally based on the SCA (Software Communications Architecture). The SCA is designed to ensure portability of waveforms across the various radios in the JTRS family. The SCA does not constrain the modem architecture, allowing the use of any combination of general-purpose processor (GPP), digital signal processor (DSP), and field programmable gate array (FPGA) devices the radio developer deems necessary within the modem to support the physical layer implementation of the target waveforms.

The SDR developed for JTRS and SCA have been developed for a specific business case and a specific user: the military. They provide a high level of reliability and security, but also at a high cost. Because of these reasons, they may not be appropriate for other applications in the commercial domain or public safety domain.
Other SDR platforms have been also created, which are not be based on SCA.

To distinguish SDR platforms not related to JTRS and to include cognitive radio capabilities, ETSI provides a new definition of these systems as Reconfigurable Radio Systems (RRS), which are defined as:
"The group of technologies for Cognitive Radio and for Software Defined Radio are all technologies for Reconfigurable Radio Systems (RRS). Such systems exploit the capabilities of reconfigurable radio and networks and self-adaptation to a dynamically changing environment, with the aim to ensure end-to-end connectivity" from [3].

Because the platform used in the research work presented here is not based on SCA, we will use the term RRS in the rest of the paper.

# 5. The DECT standard

The DECT protocol was initially released by the ETSI in October 1992.
DECT now stands for Digital Enhanced Cordless Telecommunications but initially was named Digital European Cordless Telecommunications. It is a very flexible protocol that works not only for cordless communications but also as a cellular standard and wireless LAN standard. Since the initial acronym was not DACT or D-US-CT the standard only took-off for cordless communications.
The Digital Enhanced Cordless Telecommunications (DECT) standard provides a general radio access technology for wireless telecommunications, which operates in the 1880 to 1900 MHz band using GFSK (BT =0.5) modulation.
DECT carriers are specified in EN 300 175-2 annex F for the whole frequency range 1 880 MHz to 1 980 MHz and 2 010 MHz to 2 025 MHz.
The most common protected spectrum allocation is 1 880 MHz to 1 900 MHz.
DECT is designed to provide access to a number of telecommunication network types including residential, PSTN and ISDN access, wireless PABX, GSM access, Wireless Local Loop, Cordless Terminal Mobility and it supports applications like voice telephony, fax, modem, E-mail, Internet and X.25.
The basic DECT standard is the multipart document Common Interface (CI) EN 300 175 (see references from [8] to [14]), which is often used in association with the Generic Access Profile (GAP) EN 300 444.
The DECT protocol stack presents a C(ontrol)-plane and a U(ser)-plane interface onto the stack. The C-plane is the interface to the control entity in the application, while the U-plane is designed for transport of data.

The Control Plane is the control interface for the application of the DECT handset and base system. Through this interface the network layer services can be used to eg. set-up calls and exchange end-to-end control information.
In a handset, this interface is used by the Control entity and User Interface entity in the handset which reads the keypad and controls the handsets display. Based on the users actions services on the C-plane are requested and indications from the C-plane are translated into information on the display. In a base system the C-plane is most likely used by an interworking unit to connect the DECT system to a (telephone)network (eg. ISDN). Remember that DECT is an access technology, it provides no network services like switching. The interworking unit will translate the signalling back and forth between the DECT stacks' network layer and the network protocol stack (eg. ISDN layer 3),

The User Plane is the data interface for the application of the DECT handset and base system. In telephony applications this is used to transfer ADPCM encoded speech
with a rate of 32 kbps. In a handset this needs to be encoded/decoded and coupled to the audiosystem, ie. speaker and microphone. Also echo cancellation plays an important role to guarantee the quality of the communication.
The complete schema of the Control Plane and User Plane is described in the following picture:

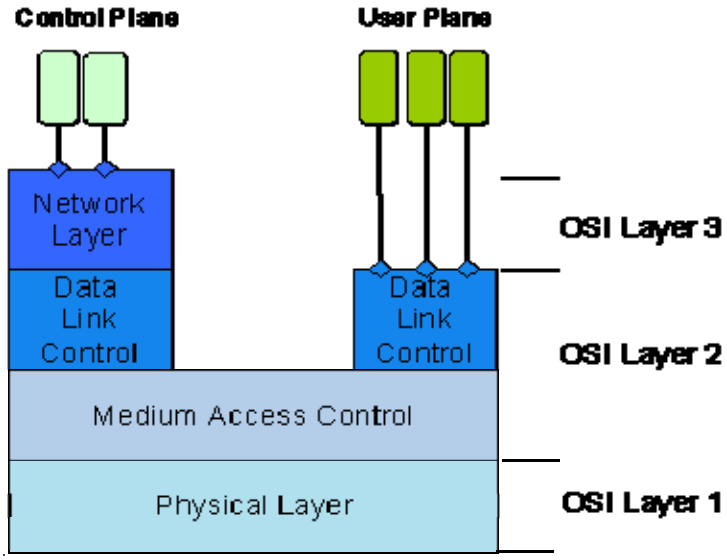



**Figure 3 Control and User Planes in DECT**

The implementation of a security attack against DECT is based on a comprehensive understanding of the various layers of the protocol stack, with specific consideration to the physical, medium access control and data link layers of the standard.

For this reason, in this section, we will describe some of the key elements of the DECT standard

The DECT is based on Time Domain Multiple Access (TDMA) The TDMA structure repeats in frames of 11 520 symbols, and the data is transmitted at a symbol rate of 1 152 ksymbol/s. Within this frame 24 full-slots are created, each consisting of two half-slots. A double slot has a length of two full slots, and starts concurrently with a full slot.

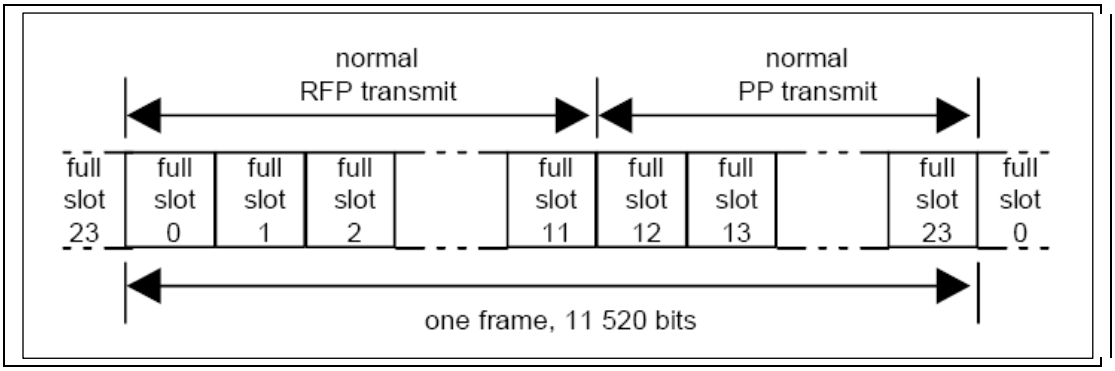



**Figure 4 DECT TMDA Structure (from reference [8])**

The internal slot structure is described in the following picture:

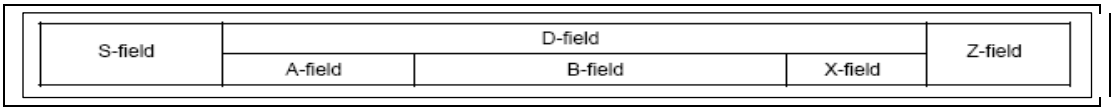



**Figure 5 DECT slot structure (from reference [9])**

The S-Field, Z-field are related to the modulation schemes used, through the following:

| Configuration | S–field | A-field | B+X+Z-field |
|---|---|---|---|
| 1a | GFSK | GFSK | GFSK |
| 1b | π/2-DBPSK | π/2-DBPSK | π/2-DBPSK |
| 2 | π/2-DBPSK | π/2-DBPSK | π/4-DQPSK |
| 3 | π/2-DBPSK | π/2-DBPSK | π/8-D8PSK |
| 4a | π/2-DBPSK | π/4-DQPSK | π/4-DQPSK |
| 4b | π/2-DBPSK | π/8-D8PSK | π/8-D8PSK |
| 5 | π/2-DBPSK | π/2-DBPSK | 16-QAM |
| 6 | π/2-DBPSK | π/2-DBPSK | 64-QAM |

**Figure 6 Relation between S-Field, Z-Field and modulation schemes in DECT (from reference [9])**

The D-field is composed by A-field, B-Field and X-field.

The following is the typical structure for a full-slot:

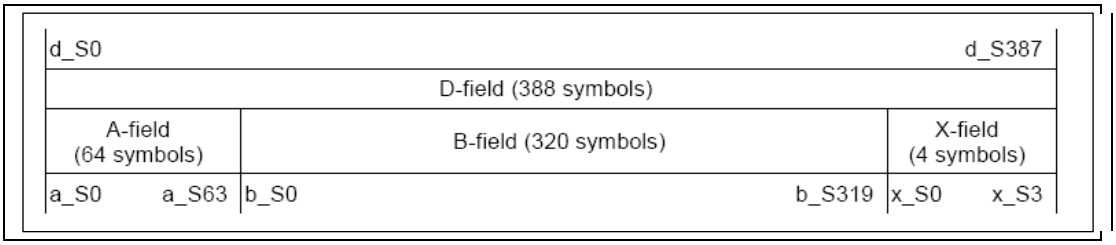



**Figure 7 DECT D-field structure (from reference [9])**

The structure of the A-field, B-field and X-field may differ depending on the modulation configuration and the size of the packet. The rule is anyway that D-field is 68 symbols + the size of the B-field.

We will now describe the most common configuration (B-field of 320 symbols).

The A-field is divided in a Header (H) of 8 bits, tail (T) varies in function of the modulation level. The remaining 16 bits are redundancy bits, RA, to provide error control on all the A-field data.

The X field is used for Cyclic Redundancy Check. The X-field consists of the last 4 bits of the B-field for 2-level modulation, the last 8 bits of the B-field for 4-level

modulation, the last 12 bits for 8-level modulation, the last 16 bits for 16-level modulation and the last 24 bits for 64-level modulation.

The type of polynomial is based on a number of factors. The main factor is the modulation configuration and it described in reference [9].

Apart from the frame structure, another important element is the knowledge of the MAC layer state machine. This provides useful information to an attacker to understand when is the best moment to implement a security attack.

The MAC layer finite state machine for the PP is:

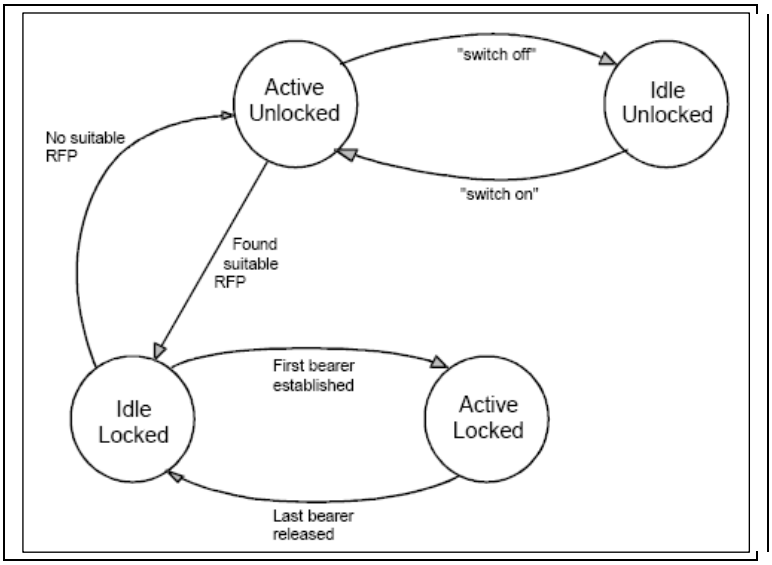



**Figure 8 The MAC layer finite state machine for the PP (from reference [9])**

A PP can exist in one of four major states at the MAC layer (from reference [9]):

1. **Active_Locked:** where the PP is synchronized to at least one RFP transmission and has one or more connections in progress.
2. **Idle_Locked:** where the PP is synchronized to at least one RFP transmission. It is able to make or receive connections, but has no connections in progress.
3. **Active_Unlocked:** where the PP is not synchronized to any RFP transmissions, and is unable to make or receive connections. The PP makes occasional attempts to detect a suitable RFP and enter the Idle_Locked state.
4. **Idle_Unlocked:** the PP is not synchronized to any RFP and does not attempt to detect RFPs.

The MAC layer finite state machine for the RFP is:

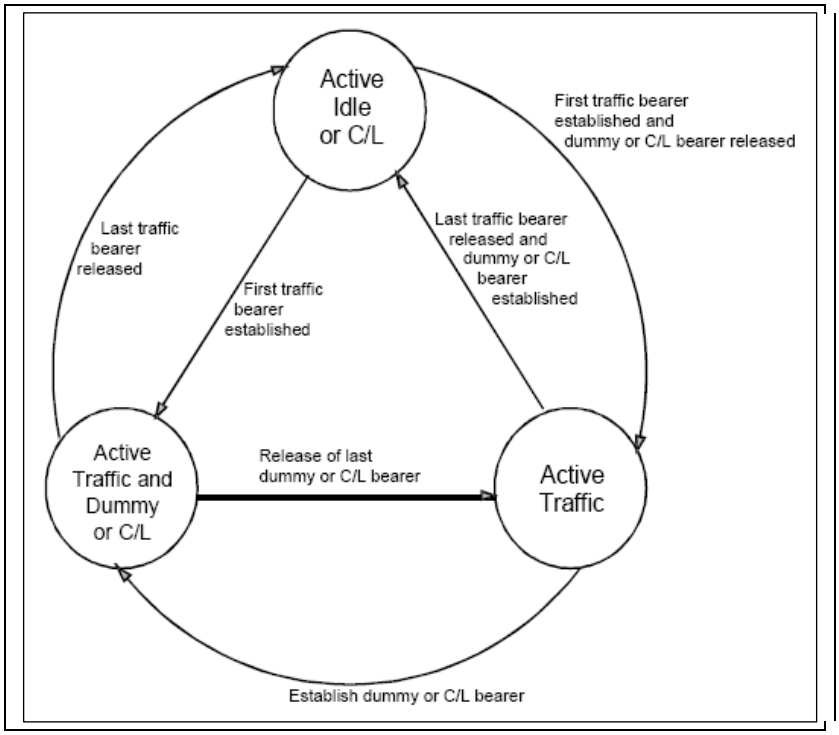



**Figure 9 DECT MAC state diagram for RFP (from reference[9])**

An RFP can exist in one of four major states at the MAC layer (from reference [9]):

1) **Inactive:** where the RFP is not receiving or transmitting.
2) **Active_Idle or C/L:** where the RFP has either at least one dummy bearer or at least one connectionless downlink bearer, and a receiver that is scanning the physical channels in a known sequence.
3) **Active_Traffic:** where the RFP has at least one traffic bearer, but does not have a dummy or a connectionless downlink bearer.
4) **Active_Traffic_and_Dummy or C/L:** where the RFP has at least one traffic bearer and is also maintaining one dummy or connectionless downlink bearer.

A number of useful information can be collected by retrieving the ARI (Access Rights Identity) number.

DECT provides a flexible radio access technology for a large variety of private and public networks or systems. This leads to different requirements on e.g. sub-system grouping, distribution and installation of equipment, identity allocations and subscription.
Therefore five access rights classes A to E and a number of IPUIs have been defined to meet the need for a differentiation in the identity structures.

| ARI class | Environment | SARI/ TARI | PARK class | IPUI type |
|---|---|---|---|---|
| A | Residential and private (PBX) single and small multiple cell systems | No | A | N, S |
| B | Private (PABXs) multiple cell | Yes | B | O, S, T |
| C | Public single- and multiple cell systems | Yes | C | P, Q, R, S, U |
| D | Public DECT access to a GSM/UMTS operator network | Yes | D | R |
| E | PP to PP direct communication (private) | Yes | E | N |

**Figure 10 Combinations of identities ARI, PARK and IPUI (from reference [12])**

ARI A class is intended to be used for small residential and private (PBX) single cell FPs and small multi-cell FPs with a maximum of 7 RFPs. This is the most commonly used by residential customers.

The RFPI A, contains a number of interesting information for a security attacker (from reference [12]):

1. **EMC**: Equipment Manufacturer's Code is allocated to each manufacturer by ETSI or by a provider authorized by ETSI. Upper limit of EMC is 65 535.
2. **FPN**: Fixed Part Number shall be allocated by the manufacturer as a unique number for each EMC. It has an upper limit of 131 071.
3. **RPN**: Radio fixed Part Number, this number is allocated by the manufacturer/installer and is used to separate a maximum of 7 different cells from each other. In case of single cell FPs, RPN = 0. This indicates for a PP that this FP does not have intercell handover, since there is only one RFP.

This information is broadcasted by the DECT terminal and base station during a communication as from the following picture:

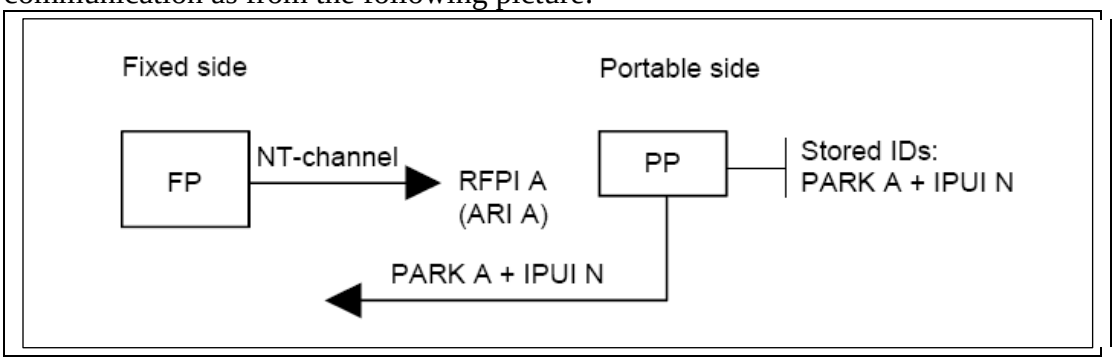



**Figure 11 Combinations of identities ARI, PARK and IPUI (from reference [12])**

A security attacker, who is able to eavesdrop the communication link and who is able to decode the lower layers and the CRC, can then extract this information and identify the passed parameters.

# 6. RRS platform

## 6.1. GNU Radio - USRP

GNU Software Radio (GSR) is an open source project that provides a free software toolkit for developing RRS running on the Linux Operating System (OS) on standard PCs (see reference [16]). While GSR is hardware-independent, it directly supports the so-called Universal Software Radio Peripheral (USRP) front end designed by Ettus et al. A top-down description of the combined GSR and USRP platform is provided in Figure 12.

The programming environment is based on an integrated runtime system composed by a signal-processing graph and signal processing blocks. The signal-processing

graph describes the data flow in the RRS and is implemented using the object-oriented scripting language Python. Signal processing blocks are functional entities implemented in C++, which operate on streams flowing from a number of input ports to a number of output ports specified per block. SWIG (Simplified Wrapper and Interface Generator) is used to create wrappers for Python around the C++ blocks.

GSR provides a large and growing software library of individual signal processing routines as well as complete signal processing blocks. The runtime system provides dynamic buffer allocation and scheduling according to fixed or dynamic I/O rates of the blocks. The scheduler supports signal graph modifications and real-time reconfigurability. The environment provides integration of the GSR with the Linux operating system to provide support for OS services like standard Linux pipeline or Inter-Process Communication (IPC). A Hardware Abstraction Layer provides support for drivers and for the management of the Hardware platform (USRP).

The USRP is a low-cost, simple and flexible peripheral, which provides both receive and transmit functionality. It is produced by Ettus Research LLC, based in Mountain View, CA, USA. Powered by a 6VDC, 3.5A power supply, it interfaces with the host computer through one Cypress FX2 USB 2.0 interface, capable of 32 Mbyte/sec . It includes one Altera Cyclone EP1C12 FPGA, connected to two Analog Devices AD9862 (each with two 12-bit 64-MSPS ADC and two 14-bit 128-MSPS DAC)

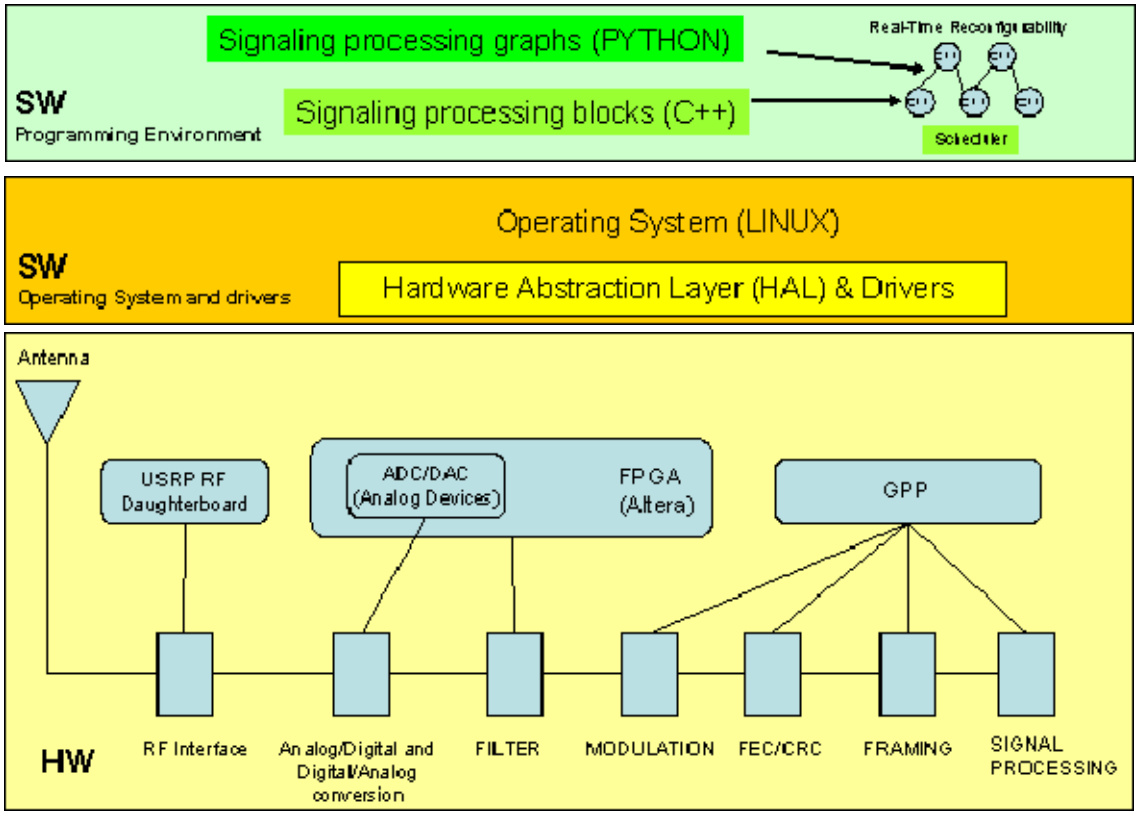



**Figure 12 GNU Radio architecture**

## 6.2. Hardware architecture

The USRP consists of one main board and up to 2 Rx and 2 Tx daughterboards. While the main board performs ADC & DAC conversion, sample rate decimation/interpolation, and interfacing, the daughterboards contain fixed RF front ends or direct interfaces to the mainboard's ADC & DAC. This configuration allows a high degree of flexibility because daughter-boards can be connected depending on the type of communications and RF spectrum usage.

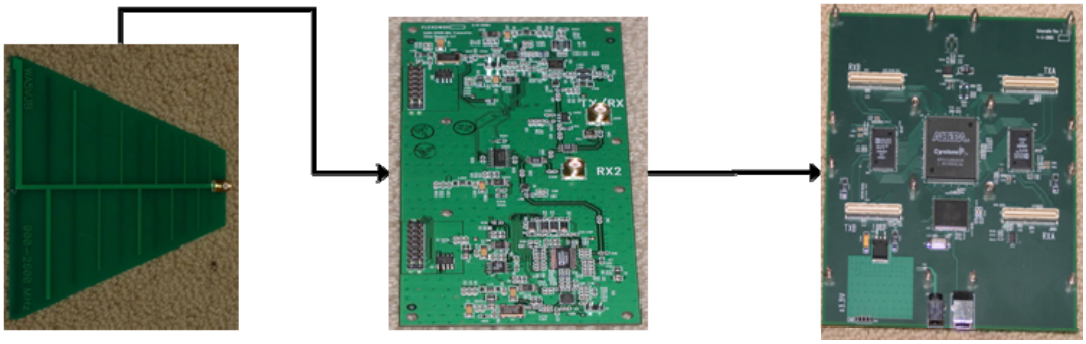



**Figure 13 Reconfigurable hardware architecture, antenna, rfx 1800 daughterboard and USRP main board**

## 6.3. The antenna and coax connection

USPR can use a number of antennas depending on the type of frequency used.
For this project, we used the LP0926 log periodic PCB antenna with frequency range from 900 MHz to 2.6 GHz and average. 5-6dBi Gain.
This antenna is appropriate for the DECT frequency range.

## 6.4. The RF front end, USRP daughterboard

The RF front end is the RFX 1800 daughterboard, which works in the DECT range of frequencies.
The RFX1800 daughterboard has a frequency range from 1.5 to 2.1 GHz, transmit Power of 100mW (20dBm) and 30 MHz transmit and receive bandwidth.

The sample rate is 64 Mbit/sec, USRP clock = 64 MHz, FPGA clock = 64 MHz and 16 digital I/O lines to control external devices like antenna switches.

The device is fully synchronized to the internal clock, however in order to improve frequency stability and phase noise an external clock source was used.

The block diagrams of the RF front end are shown in the following picture:

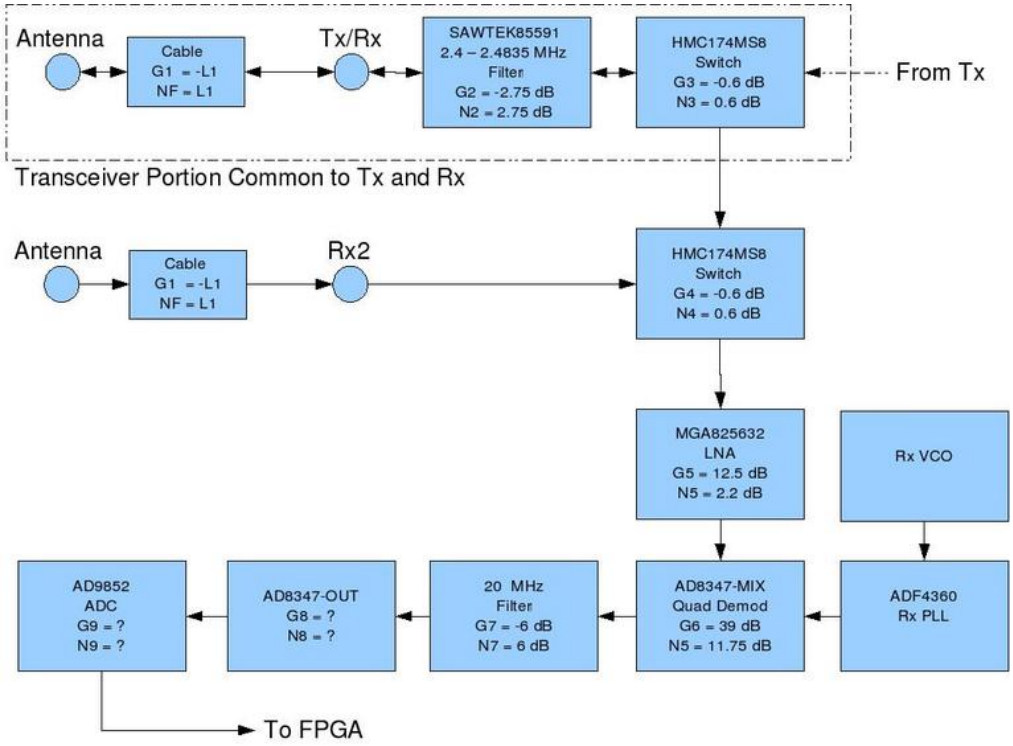



**Figure 14 Block diagram of RFX1800 transmit signal path:**

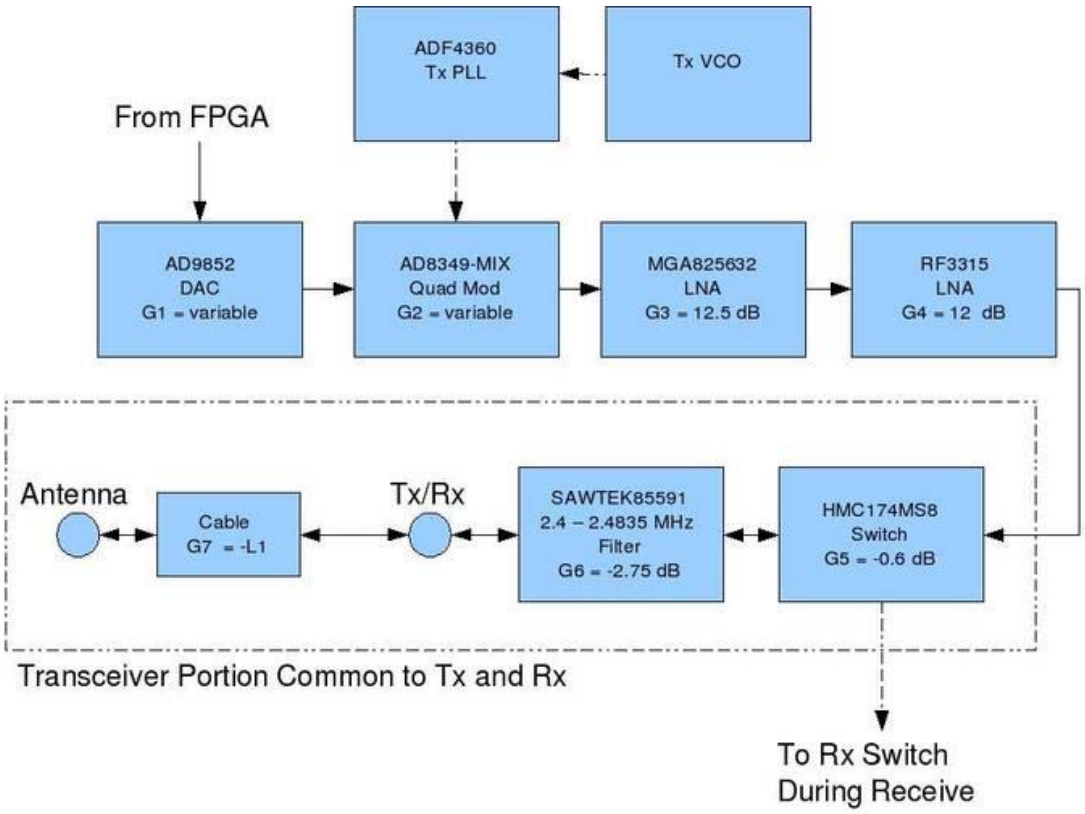



**Figure 15 Block diagram of RFX1800 receive signal path**

## 6.5. The digitizer ADC/DAC section and digital up/down conversion: the USRP main board

ADC/DAC inside USRP implements sampling and quantization functionality
The analog interface portion contains four analog to digital converters (ADC) and four digital to analog convertors (DAC). The ADC's operate at 64 million samples per second (Msps) and the DAC's operate at 128 Msps. Since the USB bus operates at a maximum rate of 480 million bits per second (Mbps), the FPGA must reduce the sample rate in the receive path and increase the sample rate in the transmit path to match the sample rates between the high speed data converter and the lower speeds supported by the USB connection.

The bottleneck of the system is in-fact the USB connection.

The ADC/DAC chip is implemented by AD9862. The AD9862 provides several functions. Each receive section contains four ADC's. Before the ADC's there are programmable gain amplifiers (PGA) available to adjust the input signal level in order to maximize use of the ADC's dynamic range. The transmit path provides an interpolator and upconverter to match the output sample rate to the DAC sample rate and convert the baseband input to a low IF output. There are PGA's after the DAC's.

Most of the receive signal processing is performed in the FPGA.
The standard FPGA firmware provides two Digital Downconverters (DDC). The FPGA uses a multiplexer to connect the input streams from each of the ADC's to the inputs of the DDC's. This multiplexer allows the USRP to support both real and complex input signals. The DDC's operate as real downconverters using the data from one ADC fed into the real channel or as complex DDC's where the data from one ADC is fed to the real channel and the data from another ADC is fed to the complex channel via the multiplexer.

The following pictures provide a description of the digital down-conversion and decimation stage, digital up-conversion stage and overall architecture for the transmit and receive signal paths in the USRP.

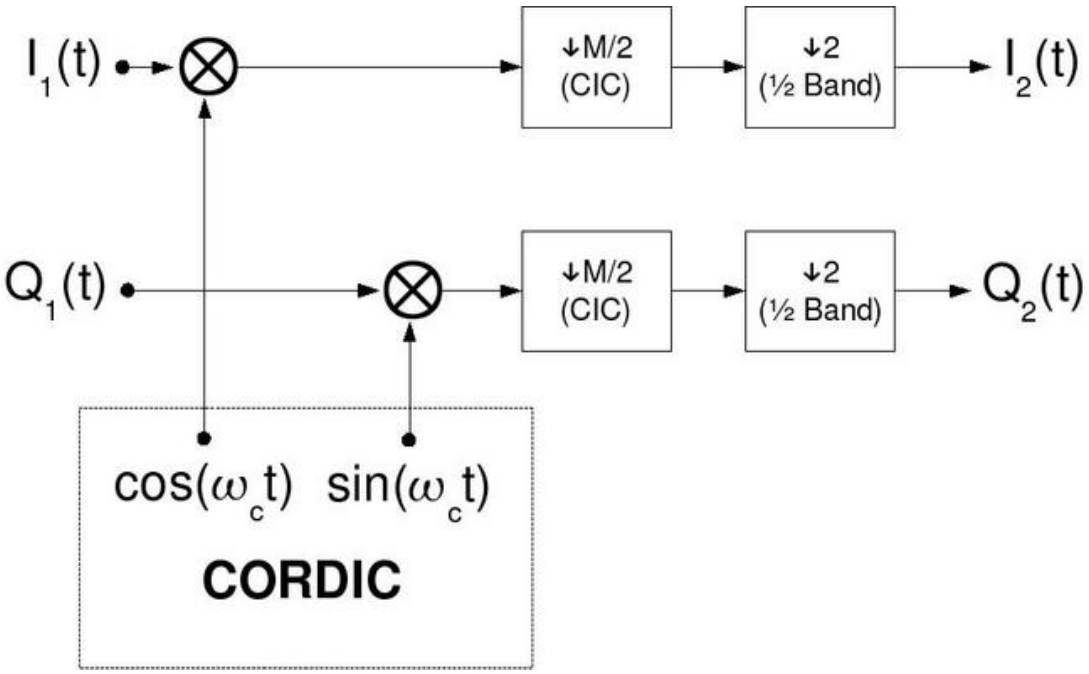



**Figure 16 Block diagram of the digital down-conversion and decimation stage:**

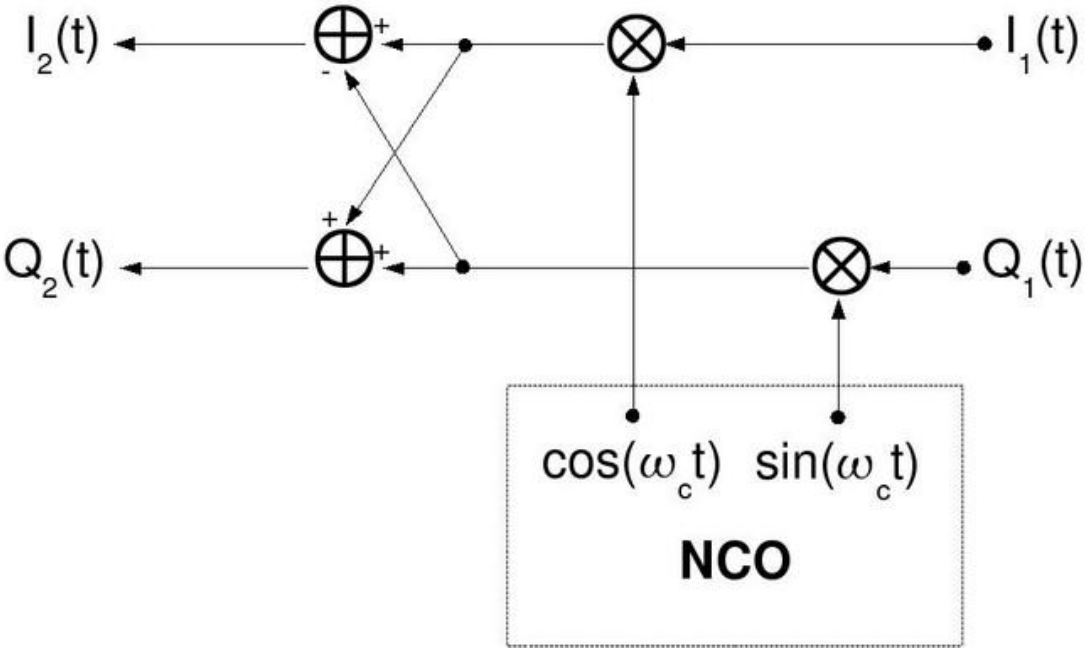



**Figure 17 Block diagram of the digital up-conversion stage.**

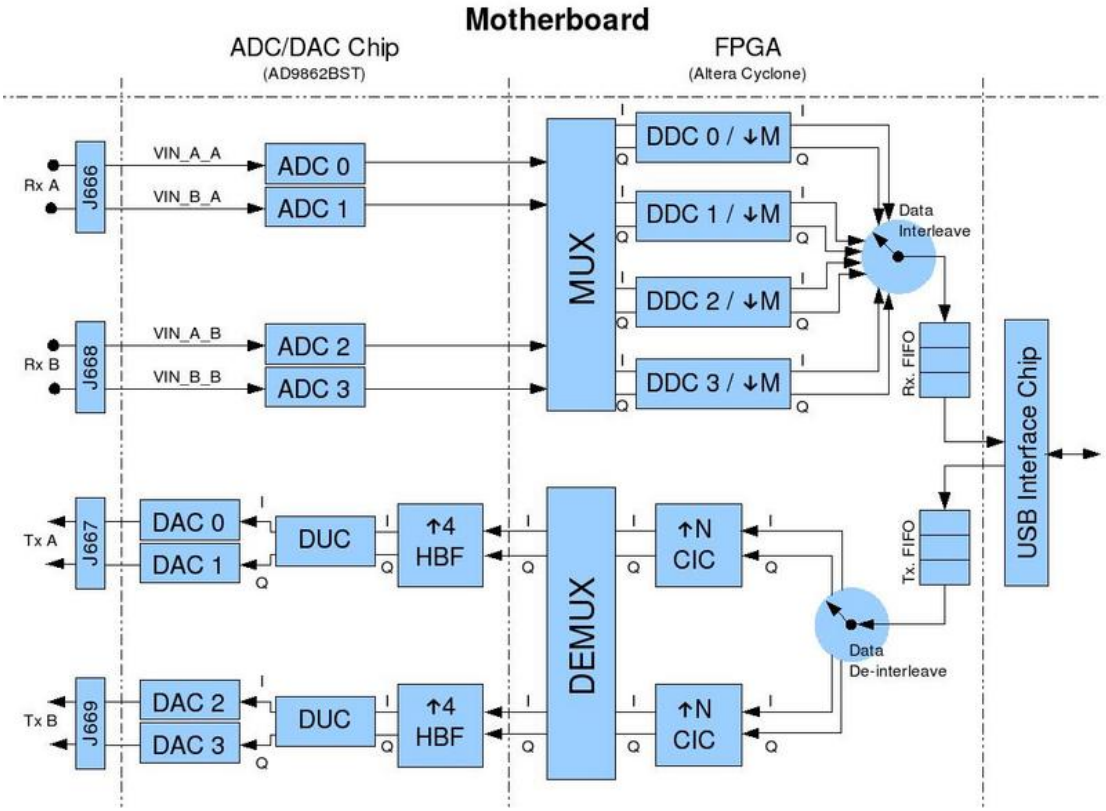



**Figure 18 block diagram of the transmit and receive signal paths in the USRP**

# 7. Status of research and reference papers

RRSs have been extensively researched as technology enabler for Cognitive Radio and Dynamic Spectrum Management, but the application of RRS as tools to implement wireless attacks is relatively recent.

Most of recent research activity exploits the reconfigurability properties of RRS systems and their flexibility to implement wireless security attacks against conventional wireless communications systems like GSM/UMTS or DECT.

In [5], the authors presented techniques for eavesdropping on Bluetooth data, therefore eliminating any confidentiality associated with packets. They show how packets can be intercepted and unwhitened. They also provide the first single channel open-source Bluetooth sniffer.

Reference [4] describes the A5 cracking project, where a GSM sniffer is implemented using a GnuRadio platform and algorithms are implemented to crack the A5 encryption algorithm of GSM. A5/1 is the strong version of the encryption algorithm used by GSM customers in Europe to protect the over-the-air privacy of their cellular voice and data communication. The A5 cracking project has the objective to use the GnuRadio platform to decode the A5/1 algorithm.

A description of the GSM sniffer is in [6], which is a website focused on the use of GnuRadio platform to implement security attacks against communication systems like GSM, UMTS and 3G.

Reference [7] describes how a RRS can be used to eavesdrop information on medical equipment like defibrillators or sensors, which can be exploited to implement attacks or implement threats to the privacy of patients. The authors partially reverse-

engineered the ICD's communications protocol with an oscilloscope and a software radio and then implemented several software radio-based attacks that could compromise patient safety and patient privacy.
In [18], the authors describe the implementation of a security attack on DECT by eavesdropping using a ComOnAir PCMCIA VoIP laptop card and a Linux computer. No RRS platform was used, but the approach is very similar to what is described in this paper.
In [20], the authors described the use of GnuRadio to implement a security attack against the RFID cards used in the Boston T subway system. The report produced by the authors was considered so dangerous that the Massachusetts Bay Transit Authority (MBTA) blocked their presentation at DEFCON conference through a legal injunction.

# 8. Prototype Implementation

## 8.1. Introduction

Modern bidirectional digital telecommunications systems are composed by an analog integrated front-end for transmission and reception of RF signals that always include in some form filtering and up-down conversion. The received signal is therefore translated to base band, digitized by ADC and processed by dedicated processors, while the transmitted signal proceeds in an inverse way from the base band processors to the DAC and so on to the antenna.

Technological progress on the digital side of these systems has opened the way to a novel implementation approach: bringing the digital signal as close as possible to the antenna(s). Using this approach, most of the signal processing is done in software rather than in hardware, thus these systems have become known as Software Defined Radios.

Using this approach, a significant subset of the digital personal communications systems DECT has been implemented using a commercially available SDR platform. This platform is composed of low-cost universal hardware front end (USRP) and an open source software stack that performs all the RF as well as the base-band processing on the digitized samples. This in turn demonstrates the capability to implement in software real-time and non-real time processing by using multiple threads and SIMD as well MISD parallel computing paradigms. This was achieved using a commercially available multi-core GPP (a commercial Linux based PC).

The software components also provide a plug-in environment where simulation code can be added or existing code modified in order to test and verify research hypothesis on a real-time, real-world platform, providing a low cost model for testing of simulation programs and software-based signal processing.

## 8.2. Dimensioning of RF parameters

**RF channel calculation**:
Fc=carrier frequency in Hz
Nc=Channel number
Europe

0<=c<=9
Fc = F0 - c · 1,728 MHz
F0 = 1 897,344 MHz.

USA

49<=c<=53
Fc = F9 + c · 1,728 MHz
F9 = 1 881,792 MHz;

**FPGA decimation**

```
Adc rate Adc= 64MSamples/sec.
Dect GAP bitrate= 1.152MBit/sec.
Dect GAP symbolrate Sy= 1.152MSymbol/sec.
```

Decim= USRP FPGA decimation rate
Baseband sampling rate Bs=Adc/(CIC*Halfband)
CIC=Decim/2
Halfband=1/2
Bs= Adc / Decim

```
Dect samples per symbol Sps= Bs/Sy
```

In order to keep the information content of the signal intact, according to Nyquist the number of samples per symbol Sps should be equal or bigger than two. The dimensioning of the decimation factor in the FPGA should be the largest integer even number that satisfies this criterion.
Case Dfpga = 24
Bs=2.67 Msamples/sec
Sps=2.318 samples/symbol (compliant)

Case Dfpga=26
Bs=2.462 Msamples/sec
Sps=2.137 samples/symbol (compliant)

Case Dfpga=28
Bs=2.285 Msamples/sec
Sps=1.984 samples/symbol (non compliant)

**Channel filter**

```
Dect GAP bitrate By= 1.152 Mbit/sec.
Dect GAP symbolrate Sy= 1.152 MSymbol/sec.
```

The channel filter used is a low-pass decimating FIR filter with 20 taps and decimation factor of 1.

```
Parameters used for taps calculation:

Gain=2.0
Passband ripple = 1.0 db
Stopband attenuation = 60 db
```

Channel bandwidth = 1,728 MHz
Occupied bandwidth = Sy * 1.03 (Gaussian filter with BT=0.5) = 1.18656 MHz

## 8.3. Dimensioning of Baseband parameters

**Demodulator**
The GnuRadio standard GMSK demodulator is a one-bit differential detector, it looks at the phase-change over one sample.
It uses non-coherent fm demodulation to extract the baseband signal. Then it estimates the signal clock that is used to re-sample the demodulated signal. A hard decision with fixed threshold is employed at the sampling instants to recover the bit values.
The input is the complex modulated signal at baseband. The output is a stream of bits packed 1 bit per byte (the LSB).
*Parameters:*
Digital Modulation type=GMSK with BT=0.5
Samples per symbol= 2.67 or 2.462 (see calculations above for Sps)
Clock recovery :
omega = 2.136752
gain mu = 0.175000
mu = 0.500000
omega rel. limit = 0.005000
frequency error = 0.000000

**Correlator**
The correlator block calculates the hamming distance between the input bit-flow and the S field of a DECT packet.
*Parameter*
Nright=2
Number of errors tolerated in the preamble in order to be recognized as valid.
Setting Nright too low leads to increased frame skipping.
Setting it too high can lead to excessive load for the framer and frame losses due to excessive sensitivity to weak unsynchronized co-channel interferers (other DECT PPs or FPs). This is useful for monitoring weak interferers but harmful when trying to lock to a single useful signal without losing frame synchronization

**Framer**
*Parameters*

Number of out of sync frames received before a new sync is forced
Number of non aligned bits tolerated before a frame is marked out of sync
Total number of bits in a frame; default 480(1 packet)*24 (number of slots in a frame)=11520

**Upper layers**
Voice coding: The voice signal has 300-3400 Hz, 4Khz bandwidth using 8KSamples/sec for a total of 64 Kb/s. StandardADPCM32 ITU G.704 vocoder is employed with uniform quantization in the GAP profile. The voice coder reduces the data rate from 64 Kb/s/ to 32 Kb/s/

Channel coding: Channel coding consists of:
- bit scrambling where the modulating signal is multiplied by a set of scrambling codes in order to avoid long sequences of zeros and ones
Framing structure:
- Error checking where the signaling part (A filed) and the payload part (B field) are protected by a number of CRC-based error checking bits, no error correction codes are used in the GAP profile.

## 8.4. Software implementation

The objective of the research is to intercept a basic unencrypted voice communication either in the uplink between PP and RFP or in the downlink between RFP and PP. These transmissions take place in the same frequency channel and are duplexed using TDD or Time Division Duplexing. For basic voice communication the first 12 frames of the DECT multi frame structure are used for the uplink and the last 12 frames are used for the downlink. Thus by tuning the receiver front-end on a live channel, the uplink and the downlink can be intercepted at once using a single real-time software receiver. The real-time synchronous software receiver reflects the structure of a traditional super-heterodyne digital packet receiver, i.e. antenna, amplifier, RF filter, tuning, IF downconverter, baseband downconverter, baseband filter, demodulator, correlator, framer; with the noteworthy difference that all blocks starting from the IF downconverter are implemented in software.

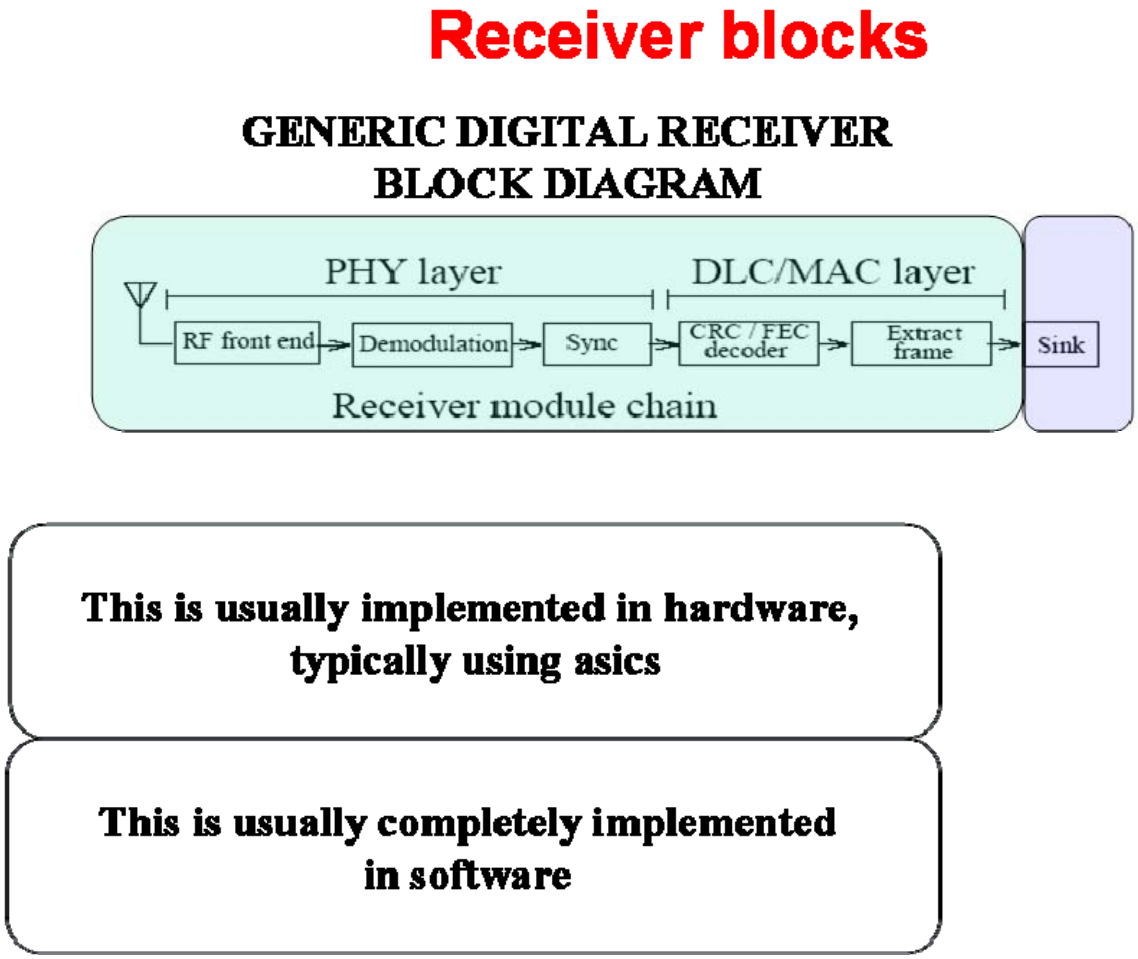



**Figure 19 Generic digital receiver block diagram**

This type of receiver alone is capable of intercepting systems like GSM. The DECT standard however includes two features that require also an asynchronous real-time computing block:

1) the presence of dynamic channel allocation
2) the absence of the frame number in the frame headers.

The SDR-based implementation of a basic DECT receiver requires at least a synchronous block for the physical layer(PHY) as well as an asynchronous block implementing a subset of the MAC layer for the control plane and the compete protocol stack for the user plane. These two software components are described in the paragraphs below.

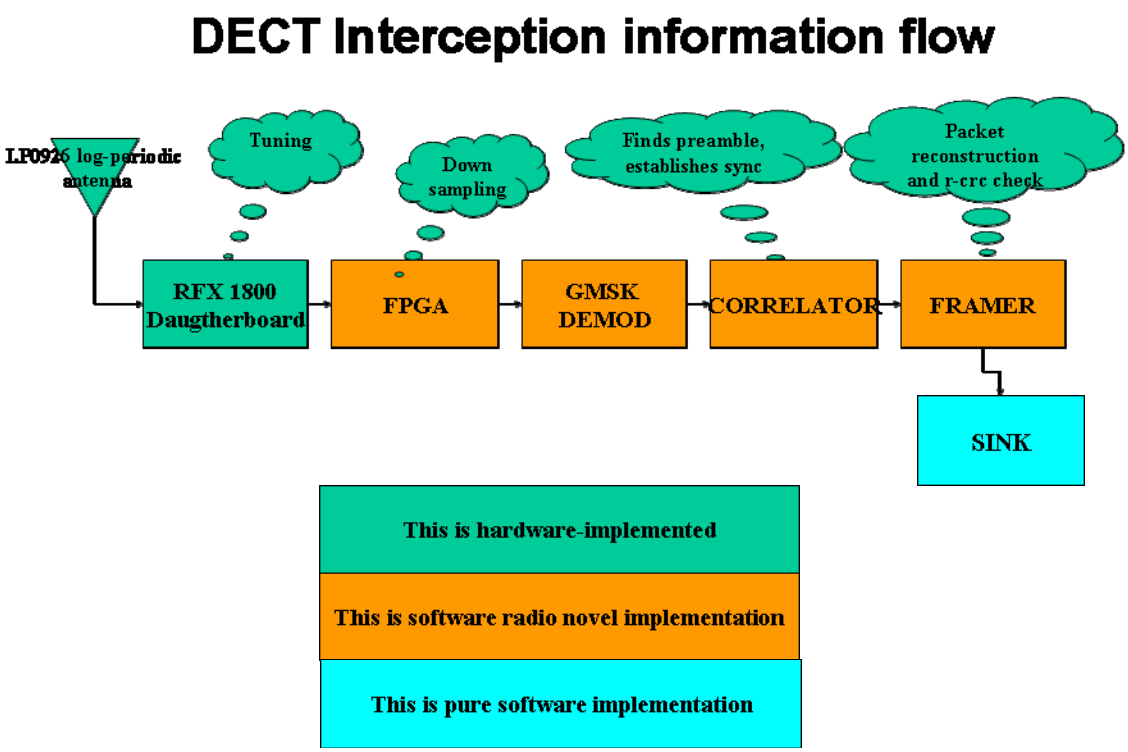



**Figure 20 Overview of the DECT software receiver**

.

### 8.4.1. Synchronous operations: real time data flow of RF samples

The real-time time software receiver is implemented using GnuRadio standard synchronous blocks that derive their timing from the USRP main clock of 64 MHz by means on FPGA digital downsampling and/or software-based dividers and multipliers

- Base band samples are sent over the USB2 line to the computer, all the subsequent blocks are fully implemented in software using a commercial PC.
- GMSK coherent demodulation implemented with GnuRadio BLKS2 blocks
- Software correlator with threshold used for identifying and extracting packet training sequence
- Software framer used for packet reconstruction and CRC decoding
- Upper layers are implemented with a C++ class that uses pthreads to spawn synchronous components as well as asynchronous one (for example finite state machines or keyboard input)

A DECT FP can have one or multiple RFP (Radio Fixed Part), one RFP can handle a single RF channel. Ordinary cordless phones have only one RFP per FP, so just a single RFP was implemented. Standard analog voice communications are just one of the capabilities of the DECT system. In this case the objective was voice interception so the Generic Access Profile (GAP) was implemented.

The GnuRadio blocks in question are all designed for synchronous, real time operations and are implemented as c++ classes running in single thread according to the GnuRadio real time scheduler using a round-robin timing sequence.

The full receiver chain is composed by: USRP->BASEBAND FILTER->GMSK DEMODULATOR->CORRELATOR->FRAMER as presented in figure Figure 20 Overview of the DECT software receiver.

The framer pushes the received packets in a msg_queue where the upper layers can pick it up, reconstruct frame and multi-frame synchronization check CRC and separate the signaling from the payload. Signaling goes o the MAC layer and stops there, while payload goes to the upper layers for descrambling and ADPCM32 decoding.
The "GMSK DEMOD" consists of a channel filter, implemented with the use of Intel built-in SIMD capability (SSE or MMX) for implementation of real-time RF filters as well as GMSK coherent demodulation and reconstruction of bit rate, symbol rate and software decimation factor.

### 8.4.2. The upper layers: asynchronous operation

Unlike other TDMA/FDMA systems, with DECT the physical layer is not sufficient to establish a one way transparent point-to-point communication data pipe as stated in the ISO standard stack.
The MAC layer has to overcome several hurdles that are standard DECT features:
1) Dynamic channel allocation -> power measurement necessary (software implementation of RSSI algorithm)
2) Slow frequency hopping-> frequency agile system, the phy layer doesn't know the frequency of the carrier a-priori
3) TDMA/TDD -> slot number used to identify bearer, again the physical layer does not know which TDMA slot will be used for the next packet and also is not aware of the starting point for frame counting.
Frame counting information maust also be passed to the upper layers in order to properly descramble the signal (scrambling sequence depends on the frame number that is not included in the frame itself but must be reconstructed)

The above-mentioned features are implemented in the MAC layer, in close communication with the PHY layer
The implementation of the DECT Media Access Control Protocol layer involves the full extraction and processing of the DECT packets. While the signaling header is handled by the MAC, the payload is checked with CRC, de scrambled and saved to disk or audio sink. A single object handles these operations. The dect_ul object is initiated by the Python waveform loader. The object creates the scoping environment and then spawns tow threads, using Pthreads before returning control of the main thread to Python. The performance of Pthreads is very well tested and documented in the Linux environment. Other operating systems supported by GnuRadio might not be able to support real-time processing. This in turn limits the waveform's portability.
The fist thread implements a subset of the DECT MAC protocol as shown in Figure 8, it also handles keyboard interaction with the user. The second thread processes the packets. The Mac thread controls the receiver and issues commands to the RF front end to change the carrier frequency and the amplifier's gain, it sends command to the PHY layers according to the API specified in the standard document. The packet thread separates signaling from payload and checks both the R-CRC for signaling and the B-CRC for payload, calculates FER as the ratio of correctly synchronized received

packets and the packets that pass all CRCs. This thread handles de-scrambling and frame synchronization, as well as voice decoding.

The requirement for time-sensitive communication of signaling and payload among layers is a very challenging aspect of reconfigurable radios. In particular the main hurdles of GnuRadio and USRP implementation are:

1) Inband signaling between blocks, requiring explicit message passing as well as implicit shared memory communications.
2) Complex multi-framing, requiring accurate synchronization and reconstruction of frame and multi-frame numbering also very important for encryption

In order to overcome these hurdles, the multi-threaded asynchronous operation of upper layers is allowed by the gr_msg_block thread-safe message passing interface. This thread-safe FIFO queue is used to communicate among the Python loader, the synchronous GnuRadio block (PHY layer) and the asynchronous upper layers threads. Shared memory (MISD) is used for inter-thread communication within the upper layer block.

### 8.4.3. Voice coding

DECT is based on 10 ms ADPCM (32 kb/s) frames.
The voice signal has 300-3400 Hz, 4Khz bandwidth using 8KSamples/sec for a total of 64 Kb/s. StandardADPCM32 ITU G.704 vocoder is employed with uniform quantization in the GAP profile.
Each 4 bit sample of ADPCM translates to 8 bit PCM sample.
ADPCM32 G.704 vocoder needed to listen to actual conversations was implemented

## 8.5. Results

The result of this research is to implement and test a RRS-based real-time DECT receiver as shown in figure Figure 21.

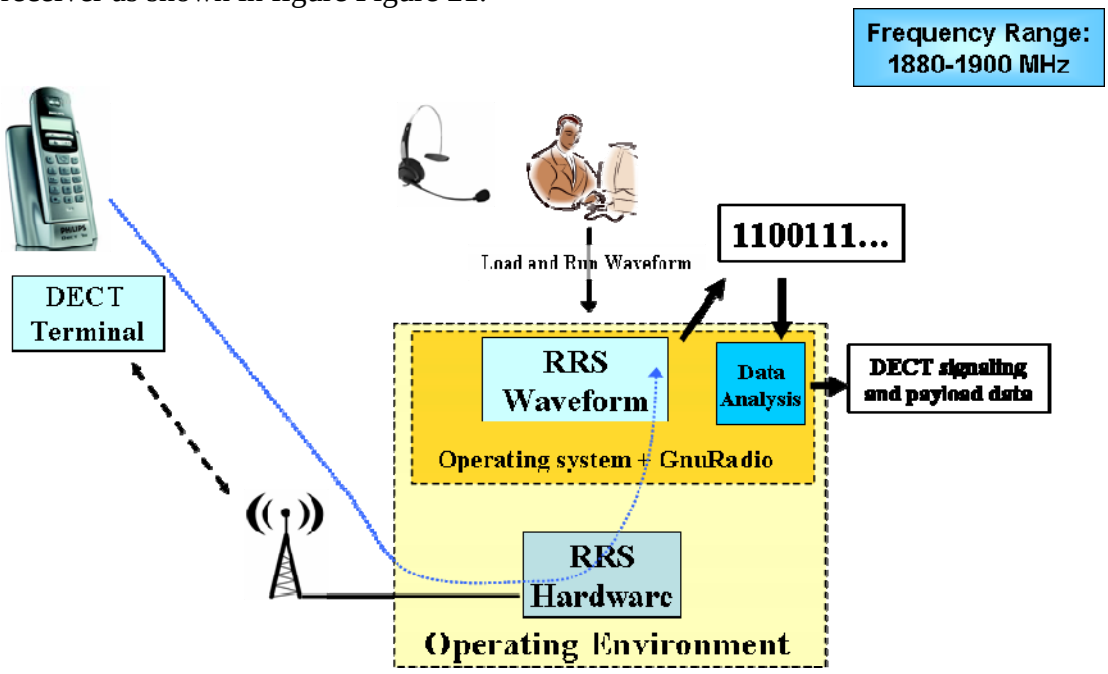



**Figure 21 DECT passive receiver (RFPI scanner and voice receiver) execution flow**

With regards to the implemented features: the physical layer and a subset of the MAC layer were implemented in the control plane, while the full user plane was implemented, limited to the standard unencrypted voice profile called GAP or Generic Access Profile.

Unlike the mainstream SDR implementations where real-time synchronous processing is implemented in the FPGA, in this work most of the processing is implemented in software, this in turn runs on an standard GPP platform (a commercial Pentium dual core PC). The RF front end is limited to the minimum achievable with current technology and even the use of FPGA reconfigurable hardware (often incorrectly considered as “software radio”) is kept to the minimum and consists of a standardized GnuRadio component (standard FPGA image) that is reusable as-is for most implementations. A further important result is the demonstration of the co-existance within the same non-real-time processor of synchronous real-time processing (filter->demodulator->correlator->framer) with real-time finite-state machine based asynchronous processing (MAC layer, error checking, voice processing).

In order to implement a DECT receiver, there is no need for expensive dedicated real-time software and hardware solutions that are a commonplace on the current RRS/SDR market. The same can be said for a whole family of telecommunication standards like GSM, GPS TETRA etc.

### 8.5.1. Measurements and validation against conventional DECT systems

The first result of this research was to identify and test a working DECT demodulator implemented using “vanilla” GnuRadio flow-graphs where synchronous processing is implemented using C++ synchronous blocks and the baseband processing is left to the python front-end that handles the waveform loading and initialization. The initial code could save to disk the baseband demodulated stream of 0s and 1s received from a real DECT cordless base station (FP) or DECT mobile phone (PP). The initial tests demonstrate that the DECT packet synchronization field (called S field) from both FP and PP can be found in the bit stream using an hex editor, the results of this test are presented below:

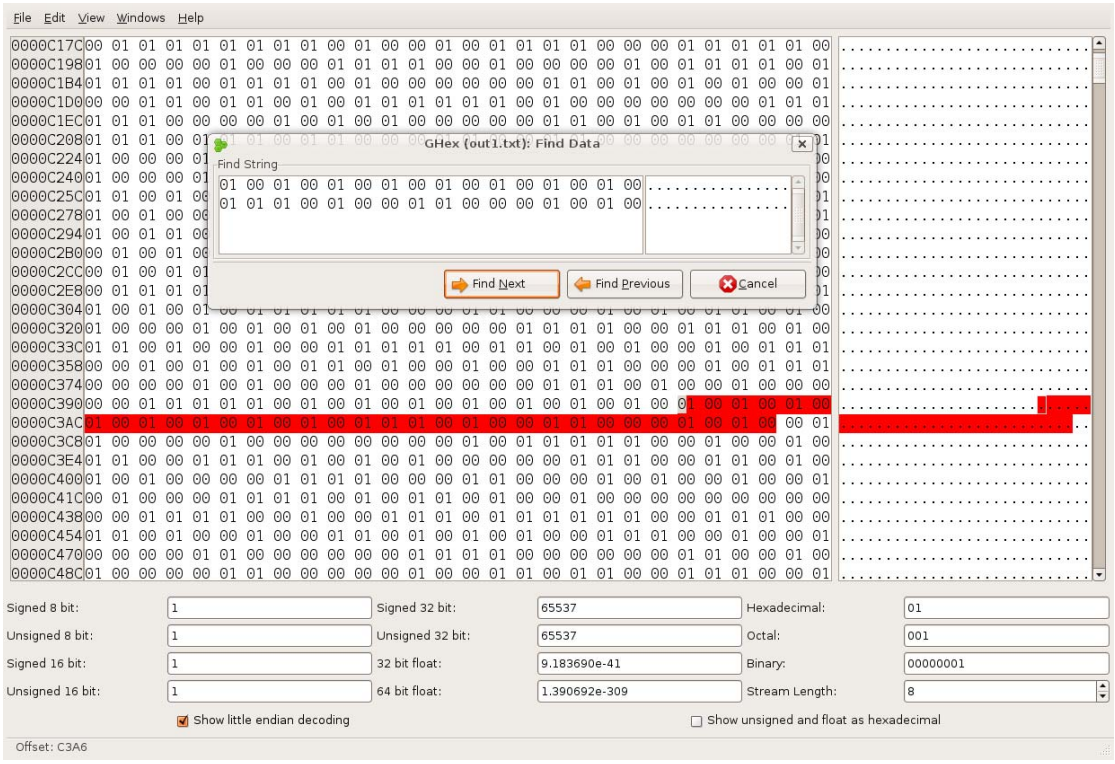



**Figure 22 S Field in the DECT bitstream.**

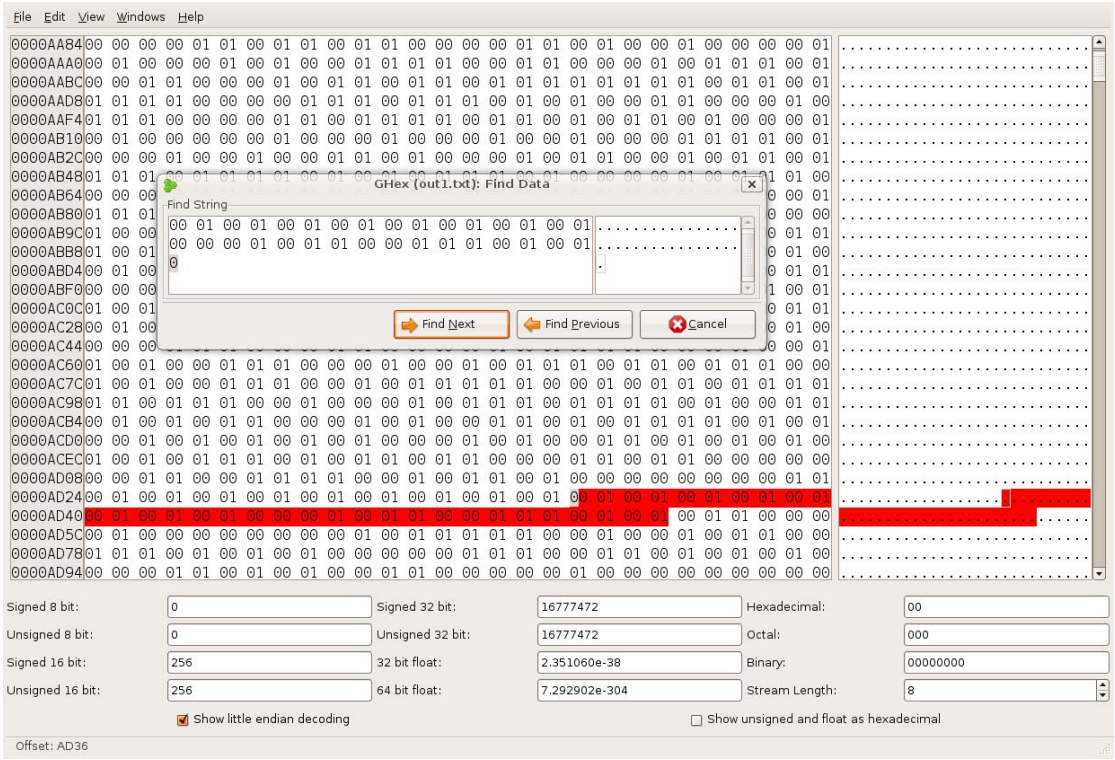



**Figure 23 S Field in the DECT bitstream.**

The next step was to test and evaluate the performance of the automated correlator and framer implemented as a C++ synchronous block described in paragraph 8.5.2. The most accurate and realistic test is to decode DECT packet headers from a number of diverse commercial DECT sets, extract the CRC validation bits as well as the PP/FP identifier called RFPID and finally check the correct reception of the header against the CRC.
Using this approach a number of useful information can be extracted for either the RFP or the PP:

- The physical RF channel used for transmission
- The received power as seen from the SDR receiver
- The performance of the reception measured with the ultimate quality indicator the Frame Erasure Rate (FER)

Results from the frequency scan conduced in the SDR lab are presented below. Two test DECT sets identified by the RFPI (big endian HEX) **BE4B3AA0209E** (Philips DECT I21) and **8C7E5F6837EF** (Bluesky BDE 3300A) are located within 30 meters from the receiver. The system is capable of finding multiple DECT bases stations within the building, outside the building and in the surroundings.

Antenna used Omni directional VERT900 824-960 MHz, 1710-1990 MHz Quad-band Cellular/PCS and ISM Band Vertical Antenna, 3dBi Gain, 9 Inches, Ideal for RFX900 and RFX1800
Only N-type identity packets were used in order to insure the proper identification of the beacon. FER is the ratio of identified S fields carrying N-type information and packets passing the R-CRC test.
The RFPID of the unknown stations have been partially blanked to protect the users.

| RFPID | RF Channel | FER | Received power linear |
|---|---|---|---|
| 604a8660???? | 4 | 1.0 | 250711.0 |
| ffa1c238???? | 2 | 1.0 | 23390.0 |
| c413ebc0???? | 9 | 1.0 | 3687.0 |
| ff8cec68???? | 2 | 0.86 | 22281.0 |
| 823e29f0???? | 5 | 1.0 | 63399.0 |
| be4b3aa0209e | 2 | 1.0 | 28480.0 |
| a2a64fa0???? | 5 | 1.0 | 46866.0 |
| 8c7e5f6837ef | 4 | 1.0 | 208524.0 |
| dcf0e6c0???? | 5 | 0.8 | 25091.0 |
| 605197b0???? | 7 | 1.0 | 31630.0 |
| #TIMESTAMP: 2009-04-17 14:42:10.952321 | | | |
| ffa1c238???? | 3 | 1.0 | 10167.0 |
| c413ebc0???? | 9 | 1.0 | 4286.0 |
| 823e29f0???? | 5 | 0.8 | 50307.0 |
| be4b3aa0209e | 2 | 1.0 | 11174.0 |
| 8c7e5f6837ef | 4 | 1.0 | 259126.0 |
| dcf0e6c0???? | 5 | 0.78 | 17736.0 |
| #TIMESTAMP: 2009-04-17 14:42:12.776178 | | | |
| 604a8660???? | 5 | 1.0 | 122603.0 |
| ffa1c238???? | 4 | 1.0 | 121702.0 |
| c413ebc0???? | 9 | 1.0 | 3467.0 |
| ff8cec68???? | 4 | 1.0 | 123143.0 |
| be4b3aa0209e | 1 | 1.0 | 13483.0 |
| 8c7e5f6837ef | 4 | 1.0 | 195725.0 |
| dcf0e6c0???? | 6 | 1.0 | 65375.0 |
| 605197b0???? | 6 | 1.0 | 46975.0 |
| #TIMESTAMP: 2009-04-17 14:42:15.056246 | | | |
| 604a8660???? | 5 | 0.75 | 36755.0 |
| ffa1c238???? | 2 | 0.9 | 7343.0 |
| c413ebc0???? | 9 | 1.0 | 2613.0 |
| ff8cec68???? | 2 | 0.89 | 8527.0 |
| be4b3aa0209e | 1 | 1.0 | 9115.0 |
| 8c7e5f6837ef | 4 | 1.0 | 125150.0 |
| dcf0e6c0???? | 5 | 0.67 | 49045.0 |
| 605197b0???? | 6 | 1.0 | 28616.0 |
| #TIMESTAMP: 2009-04-17 14:42:16.952277 | | | |
| ffa1c238???? | 3 | 1.0 | 1623.0 |
| c413ebc0???? | 9 | 1.0 | 5420.0 |
| ff8cec68???? | 3 | 1.0 | 1501.0 |
| be4b3aa0209e | 1 | 1.0 | 15746.0 |
| 8c7e5f6837ef | 4 | 1.0 | 139460.0 |
| dcf0e6c0???? | 5 | 1.0 | 63627.0 |
| #TIMESTAMP: 2009-04-17 14:42:19.232405 | | | |
| a2a64fa0 | 1 | 0.875 | 17313.0 |
| ffa1c238 | 2 | 1.0 | 46907.0 |
| c413ebc0 | 9 | 1.0 | 3141.0 |
| 8c7e5f6837ef | 4 | 1.0 | 179940.0 |
| be4b3aa0209e | 1 | 0.92 | 16064.0 |
| #TIMESTAMP: 2009-04-17 14:42:21.144145 | | | |

**Figure 24 RFPID of DECT stations**

The system can also monitor the activity of PP and produce a time-stamped report of cordless phone activity in the surroundings. This can be used for passive user monitoring or forensic research to demonstrate the online presence of a PP and its electrical distance from the RRS receiver. An example of this monitoring, using two commercial DECT sets identified by the RFPI `(big endian HEX)` **`BE4B3AA0209E`** (Philips DECT I21 using standard encryption) `and` **`8C7E5F6837EF`** (Bluesky BDE 3300A does not encrypt by default) is presented in the table below:

```
DECT PP scanning,JRC Ispra 24-06-2008 building 72
Using RX d'board B: Flex 1800 Rx MIMO B
Rx gain:          90
modulation:       gmsk_demod
bitrate:          1.152Mb/s
samples/symbol:   2.31481481481
decim:             24
Rx Frequency:     1.88173G
# 2008-06-24 15:29:47.827025
.....................................…
# 2008-06-24 15:33:19.687019
PP identifier=be4b3aa0 4   1.0   18482991.1883
# 2008-06-24 15:33:22.323079
PP identifier=be4b3aa0 3   1.0   34557833.2507
# 2008-06-24 15:33:23.731070
PP identifier=be4b3aa0 4   1.0   23710590.792
# 2008-06-24 15:33:25.083074
PP identifier=be4b3aa0 3   1.0   38254139.9621
# 2008-06-24 15:33:26.507092
PP identifier=be4b3aa0 4   1.0   6113198.83223
# 2008-06-24 15:33:28.099100
PP identifier=be4b3aa0 3   1.0   35706689.7135
# 2008-06-24 15:33:29.551085
PP identifier=be4b3aa0 3   1.0   36181927.0959
# 2008-06-24 15:33:31.007064
PP identifier=be4b3aa0 4   1.0   21657148.9441
# 2008-06-24 15:33:32.363053
.....................................…
# 2008-06-24 15:46:47.511021
PP identifier=2e48e120 1   1.0   2658159.09135
# 2008-06-24 15:46:48.831073
.....................................
# 2008-06-24 15:57:56.603021
PP identifier=8c7e5f68 4   1.0   33891083.7448
# 2008-06-24 15:57:58.211070
PP identifier=8c7e5f68 5   1.0   20149640.5341
# 2008-06-24 15:57:59.775073
PP identifier=8c7e5f68 4   1.0   36420353.1608
# 2008-06-24 15:58:01.287073
PP identifier=8c7e5f68 4   1.0   34785293.3262
# 2008-06-24 15:58:02.667074
PP identifier=8c7e5f68 5   1.0   19274289.1039
# 2008-06-24 15:58:03.987072
PP identifier=8c7e5f68 4   1.0   28470784.132
# 2008-06-24 15:58:05.323085
PP identifier=8c7e5f68 4   1.0   34081849.4537
# 2008-06-24 15:58:06.827074
PP identifier=8c7e5f68 4   1.0   29144586.9817
# 2008-06-24 15:58:08.179071
PP identifier=8c7e5f68 5   1.0   19732027.6871
# 2008-06-24 15:58:09.907073
PP identifier=8c7e5f68 4   1.0   34257220.1981
# 2008-06-24 15:58:11.287071
PP identifier=8c7e5f68 5   1.0   22498364.4268
# 2008-06-24 15:58:12.667072
# 2008-06-24 15:58:13.987019
PP identifier=8c7e5f68 3   1.0   19792485.0679
# 2008-06-24 15:58:15.335074
PP identifier=8c7e5f68 2   1.0   26747298.8648
# 2008-06-24 15:58:16.707071
```

```
.....................................
# 2008-06-24 16:28:33.271017
PP identifier=604a8660 5   1.0   13029401.014
# 2008-06-24 16:28:40.227072
PP identifier=604a8660 4   1.0   19100816.9232
# 2008-06-24 16:28:41.575070
PP identifier=604a8660 4   1.0   17766922.6784
# 2008-06-24 16:28:42.907070
PP identifier=604a8660 4   1.0   21378777.1861
# 2008-06-24 16:28:44.499073
PP identifier=604a8660 5   1.0   12836767.8373
# 2008-06-24 16:28:45.875069
PP identifier=604a8660 5   1.0   12498043.606
# 2008-06-24 16:28:47.259070
PP identifier=604a8660 4   1.0   19978372.7983
# 2008-06-24 16:28:48.643072
PP identifier=604a8660 4   1.0   20935864.0163
# 2008-06-24 16:28:50.047069
PP identifier=604a8660 4   1.0   21831021.3906
# 2008-06-24 16:28:51.439070
PP identifier=604a8660 5   1.0   11597056.9136
# 2008-06-24 16:28:52.771071
PP identifier=604a8660 4   1.0   21968357.9708
# 2008-06-24 16:28:54.195070
PP identifier=604a8660 4   1.0   20525910.4907
# 2008-06-24 16:28:55.547068
PP identifier=604a8660 4   1.0   18217326.793
# 2008-06-24 16:28:56.867072
PP identifier=604a8660 4   1.0   17010647.4658
# 2008-06-24 16:28:58.187068
PP identifier=604a8660 5   1.0   13852239.6882
# 2008-06-24 16:28:59.659070
PP identifier=604a8660 4   1.0   18397298.559
# 2008-06-24 16:29:01.047071
PP identifier=604a8660 4   1.0   16371961.9419
# 2008-06-24 16:29:02.379072
PP identifier=604a8660 5   1.0   10756888.3341
# 2008-06-24 16:29:03.699070
PP identifier=604a8660 4   1.0   19050819.1432
# 2008-06-24 16:29:05.027071
PP identifier=604a8660 5   1.0   13401687.6166
# 2008-06-24 16:29:06.463071
PP identifier=604a8660 4   1.0   16923361.6692
# 2008-06-24 16:29:07.783069
# 2008-06-24 16:29:09.103020
.....................................
2008-06-24 17:10:28.291019
```
Total scanning time 1 hour, 19 minutes.

**Figure 25 Time-stamped report of cordless phone activity**

The receiver was further refined in order to decode record and/or live play back the voice payload found on the unprotected unencrypted B fields of DECT terminals/base stations in GAP mode thus implementing a limited but fully functional subset of the upper layers of the DECT protocol's OSI stack.

An audio sample recorded from the DECT RFP numbered **8C7E5F6837EF** (Bluesky BDE 3300A) is presented below. For comparison the same sample was recorded

directly from the PP using a microphone. The two tracks are presented in the figure below for comparison. The audio signal was a series of bitonal signals from the JRC PBX, starting with a fast busy high frequency tone and moving to a slow Italian-type intermittent dialtone. The timing of the two signals are consistent, however the harmonic content is distorted in the second sample due to loss of synchronization by the RRS-implemented framer and the distortion introduced by the ADPCM32 voice decoder, probably due to a buggy/incorrect implementation of the ITU G.704 standard.

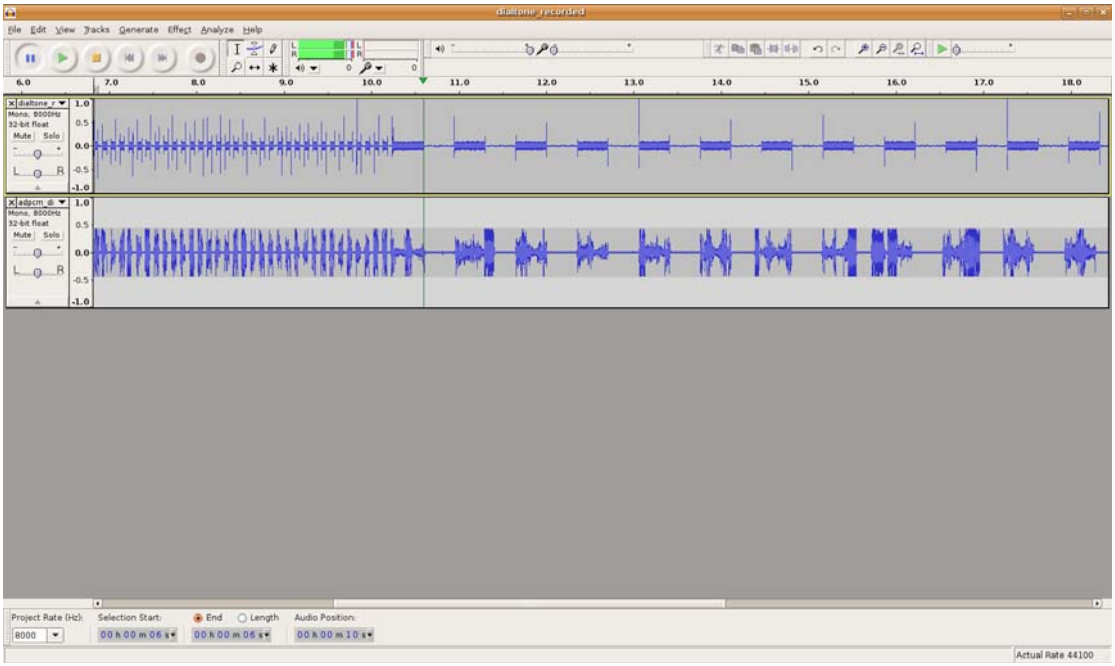


**Figure 26 Recording of DECT audio sample through GnuRadio**

Plug in simulation code can be inserted in the stack at any point, both in the real time side of the flow graph (Physical layer) using GnuRadio standard sync-blocks and in the upper asynchronous layers, using c++ APIs, ptherads and GnuRadio msg_blocks for thread-safe message-passing and synchronization.

The waveform that was implemented and tested offers additional benefits for research. The raw signals can be extracted and processed/saved at any stage of the signal path. Moreover the standardized interface between processing blocks is a great benefit since it permits a great flexibility for improvement/replacement of these blocks, creating a real-time plug-in environment for both synchronous and asynchronous signal blocks.

The following points were addressed in order to achieve this result:

- fully implemented framing and multi-framing synchronization.
- Upper layers were implemented in order to rebuild channel coding and multiplexes.
- Descrambling, CRC checking and software interface for Plug-in for cryptography were implemented

### 8.5.2. Plug-in interface for real-time synchronous modules (PHY)

The software interface implements GnuRadio real time blocks called hier-blocks by inheriting its properties from C++ objects. in this specific application byte format (unsigned char or octets) in small endian is used for input and output, note that the incoming data rate must be exactly equal to the outgoing one, in order to avoid USRP underflows or overflows. Each byte exchanged contains as its most significant bit, a single bit of payload. All the other bits are either unused or used as inter-block signaling. Any signaling block conforming to the GnuRadio hier block 2 standard and with the above provisions can be plugged in the signal flow. Any intermediate signal can be extracted duplicated or injected.

### 8.5.3. Plug-in interface requirements for upper layers

This paragraph describe the plug-in interface requirements for MAC, DLC and user plane.
A single class called dect_ul implements the upper layers, starting from the MAC layer.
The asynchronous communication between blocks is handled by a msg_queue, a simple thread-safe message passing queue the can be managed in both blocking and non-blocking mode. The queue carries a struct data type that includes a float, 2 integers, and an arbitrary-length generic data pointer. The former can be used for signaling while the latter is used for payload.
The Python code that acts as the command-line options parser, the FPGA image loader, the waveform loader, the objects instantiator, the queue starter and the program threads spawner. It also enables addressing for all of the waveform objects and devices.
Thus in order to add a plug-in asynchronous block it must be written in C++ or Python, initiated from the main Python code and provided with one ore more msg_queues for payload and signaling I/O.

## 8.6. Further research

### 8.6.1. Base station and mobile emulation

The DECT standard uses an FDD dulpexing. With a flexible all-software implementation of the protocol stack it is possible to build a receiver that behaves as an FP and a PP at the same time. Thus it is possible to intercept uplink and downlink at the same time or to implement direct push to talk communications.
TDMA frame and multi-frame synchronization with commercially available cordless base stations and handset was achieved however the current implementation is limited to passive reception of DECT signals,
The future implementation of an integrated transmission chain will enable the system to act as a repeater or cell extender. Another interesting possibility would be to use the system as a test-bed for evaluating man-in-the-middle attack strategies.

### 8.6.2. Weak interferers detection

The capability to identify RFPID sequences was extended to co-channel interferers that are likely unaware of each other, paving the way for implementation of improved

frequency scanning and detect-and-avoid techniques that can form the base of a cognitive radio test-bed. Monitoring and interception capabilities can greatly benefit from the capability of locking on weak interferers. The possibility of modifying the PHY layer is unique to the full-software implementation and was limited before to high-end test equipment and instrumentation.

# 9. Table of figures:

# 10. Bibliography


[1]. Briefing Paper. SOFTWARE DEFINED RADIO, Directorate-General for External Policies of the Union. http://www.europarl.europa.eu/activities/expert/eStudies.do?languageEN. Last Accessed 19-01-2009.

[2]. The Joint Tactical Radio System (JTRS) and the Army's Future Combat System (FCS): Issues for Congress. CRS Report for Congress.

[3]. European Telecommunications Standards Institute (ETSI). http://www.etsi.org/WebSite/technologies/RRS.aspx..

[4]. J. Lackey and D. Hulton. The A5 cracking project: Practical attacks on GSM using GNU radio and FPGAs. In Chaos Communication Camp, 2007.

[5]. D. Spill and A. Bittau. BlueSniff: Eve meets Alice and Bluetooth. In Proceedings of USENIX Workshop on Offensive Technologies (WOOT), August 2007.

[6]. The GSM Software Project. http://wiki.thc.org/gsm.

[7]. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. Daniel Halperin, University of Washington, Thomas S. Heydt-Benjamin, University of Massachusetts Amherst and Benjamin Ransford of University of Massachusetts Amherst. 2008 IEEE Symposium on Security and Privacy.

[8]. ETSI EN 300 175-2: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 2: Physical Layer (PHL)".

[9]. ETSI EN 300 175-3: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 3: Medium Access Control (MAC) layer".

[10]. ETSI EN 300 175-4: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 4: Data Link Control (DLC) layer".

[11]. ETSI EN 300 175-5: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 5: Network (NWK) layer".

[12]. ETSI EN 300 175-6: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 6: Identities and addressing".

[13]. ETSI EN 300 175-7: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 7: Security features".

[14]. ETSI EN 300 175-8: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 8: Speech coding and transmission".

[15]. ETSI EN 300 176 (all parts): "Digital Enhanced Cordless Telecommunications (DECT); Test specification".

[16]. Eric Blossom, Exploring GNU Radio. Last Accessed 16 September 2008. http://www.gnu.org/software/GnuRadio/doc/exploring-GnuRadio.html.

[17]. The OpenBTS Project. Supporting integrated mac and phy software development for the usrp sdr.

[18]. DECT Security threats. http://www.h-online.com/security/25C3-Serious-security-vulnerabilities-in-DECT-wireless-telephony--/news/112326. Last accessed 23 February 2009.

[19]. The goal of deDECTed.org is to understand DECT and DECT security better and to create an Open Source implementation of the DECT standard. https://dedected.org/trac. Last accessed 23 February 2009.

[20]. Anatomy of a subway hack by Russell Ryan, Zack Anderson, Alessandro Chiesa. http://tech.mit.edu/V128/N30/subway/Defcon_Presentation.pdf. Last Accessed 16/06/2009.

European Commission

**EUR 23963 EN – Joint Research Centre – Institute for the Protection and Security of the Citizen**
Reconfigurable Radio System Test bed for security research
Authors: Raimondo Giuliani, Gianmarco Baldini, Dimitrios Symeonidis
Luxembourg: Office for Official Publications of the European Communities
2009 – 43 pp. – 21 x 29.7 cm
EUR – Scientific and Technical Research series – ISSN 1018-5593

**Abstract**
Technological progress on the digital processing has opened the way to a novel implementation approach for wireless communication platforms where most of the digital signal processing is done in software rather than in hardware. Such systems have been known as Software Defined Radio (SDR) or Reconfigurable Radio Systems (RRS).

A typical SDR/RRS is able to execute all the radio frequency and base-band processing though software components rather then hardware components as in conventional radio communication systems. This capability provides a high level of reconfigurability and the possibility to implement a number of different algorithms for digital processing. Therefore, SDR/RRS can be used for a variety or purposes including the possibility of implementing wireless security attacks against conventional communication systems.

In this technical report, we present an application of the SDR/RRS platform to implement a security attack against a DECT platform. The SDR/RRS platform has been used to implement a DECT demodulator and a processing module to eavesdrop and capture user and control data transmitted by a DECT system. The commercially available Universal Software Radio Peripheral (USRP) has been used as SDR/RRS platform for the development of the prototype.

The paper presents the technical challenges and implementation details in the development of the prototype and an overview of the capabilities of the USRP to implement wireless security attacks. The SDR/RRS platform used in the project is quite versatile and it can be used for a number of other applications related to DECT or other wireless communication systems.

The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.