



Integrated Project - EUWB

Contract No 215669

Deliverable

D7.1.2b

**(V)HDR platform requirements, feasibility analysis and specification
(final)**

Contractual date:	M18
Actual date:	M19
Authors:	Amir Krause, Alexander Weir, Axel Schmidt, Dirk Pannicke, Yue Chen, Xu Dong, Shangbo Wang, Christian Kocks, Alexander Viessmann, Guido H. Bruck, Peter Jung
Participants:	WIS, TESUK, GWT, HTW, UDE
Work package:	WP7
Security:	PU
Nature:	Report
Version:	1.0
Total number of pages:	65

Abstract

This document is the final version of Deliverable D7.1.2b to be delivered at M18. It provides an overview of the final version of the (V)HDR ‘Open’ platform, and provides a comprehensive description of the platform, including the physical layer, MAC layer, and MAC layer interface to the application layers.

Keywords

High Data Rate, WiMedia, Detect And Avoid, Physical Layer, MAC layer, MAC Layer interface.

Table of Contents

Abstract	1
Keywords	1
List of Figures	5
List of Tables	6
Abbreviations	7
1 Executive summary.....	9
2 Introduction.....	10
3 HDR “Open Platform” Specification.....	11
3.1 High-level Structure.....	11
3.2 Implementation phases plan.....	12
3.3 Integration plan.....	12
3.4 General description.....	14
3.4.1 PHY General Description.....	14
3.4.2 PHY layer partitioning.....	15
3.4.3 PHY frame structure.....	16
3.4.4 MAC General Description.....	19
3.5 Description of PHY Module.....	20
3.5.1 Main PHY & RF Characteristics.....	20
3.5.2 Additional PHY & RF features.....	28
3.5.3 PHY MAC Interface.....	28
3.6 Description of MAC Layer.....	29
3.6.1 Development Accessories.....	29
3.6.2 MAC Core Overview.....	31
3.6.3 MAC Core Interface Description.....	34
3.6.4 Application Interface Description.....	57

4 Alternative HDR-platforms 58

 4.1 Staccato platform..... 59

 4.1.1 Development support: 60

 4.2 IOGEAR’s Wireless USB-solution 61

5 Conclusions 62

Appendix I –Specification Details “TBD” List..... 63

References 64

Acknowledgement..... 65

Page intentionally left blank

List of Figures

Figure 1: Modular Approach to HDR “Open Platform” (Prototype)	10
Figure 2: Overall layer structure of the HDR “Open Platform”	11
Figure 3: Physical layer partitioning	15
Figure 4: Standard PPDU structure	16
Figure 5: Block diagram of PLCP header construction, from [4]	17
Figure 6: Encoding process for the scrambled, Reed-Solomon encoded PLCP header, from [4].....	18
Figure 7: Block diagram of PSDU construction, from [4]	18
Figure 8: Block diagram of the encoding process for the scrambled PSDU, from [4].....	19
Figure 9: WiMedia Frequency bands plan	20
Figure 10: Xilinx ML507 FPGA Development Platform component side	29
Figure 11: ML507 Board block diagram.....	30
Figure 12: MAC Digital Hardware block diagram.....	32
Figure 13: Overview of HMAC Core.....	33
Figure 14: Measurement Results Set [4]	53
Figure 15 Staccato Communications Ripcord® Development Kit	59
Figure 16: Block diagram of Staccato’s single-chip-solution SC4501 [7].....	60
Figure 17: Layer model for Staccatos's multi-protocol-support [7]	60
Figure 18: IOGEAR's USB host adapter GUWA100U and USB hub GUWH104 [8]	61

List of Tables

Table 1: Terms and Abbreviations	8
Table 2: PHY HW Features.....	12
Table 3: Revised Integration Plan	13
Table 4: WiMedia Band Plan	21
Table 5: WiMedia TFC	22
Table 6: WiMedia Data Rates	23
Table 7: OFDM symbol mapping	24
Table 8: OFDM Frame parameters	26
Table 9: Transmit Power Control.....	27
Table 10 MAC Core High-level Features Supported by HDR Open Platform.....	34
Table 11: Range Supported Field.....	53
Table 12 – TBD list.....	63

Abbreviations

AES	Advanced Encryption Standard
ASIE	Application Specific Information Element
BP	Beacon Period
BPST	Beacon Period Start
CTI	Chekmate2 Test Interface
DAA	Detect And Avoid
DCM	Dual Carrier Modulation
DME	Device Management Entity
DRP	Distributed Reservation Protocol
EUWB	CoExisting Short Range Radio by Advanced Ultra-WideBand Radio Technology
FFC	Fixed Frequency Code
FFI	Fixed Frequency Interleaving
HDR	High Data Rate (50Mbps to 480Mbps PHY bit rate)
HW	Hardware
IP	Internet Protocol
KMDF	Kernel-Mode Driver Framework
LDR	Low Data Rate
LMAC	Lower MAC Software component
MAC	Medium Access Controller
MAC-SAP	MAC Service Access Point
MAS	Medium Access Slot
MIB	Management Information Base
MIPS	Millions of instructions per seconds, a number of instructions per second for a processor can be derived by multiplying the instructions per clock and the clock speed
MLME-SAP	MAC Layer Management Entity Service Access Point
MMI	Man Machine Interface
MSDU	MAC service data unit
NDIS	Network Driver Interface Specification
OFDM	Orthogonal frequency division multiplexing
OMG	Object Management Group
PAL	Protocol Abstraction Layer

PCA	Prioritised Contention Access
PCIDRV	PCI driver
PHY	Physical layer
PLCP	Physical Layer Convergence Protocol
QPSK	Quadrature phase shift keying
RF	Radio Frequency
RTOS	Real-Time Operating System
SAP	Service Access Point
SDIO	Secure Digital Input Output
TFC	Time Frequency Codes
TFI	Time Frequency Interleaving
TPC	Transmit Power Control
UMAC	Upper MAC Software component
UML	[The] Unified Modelling Language
USB	Universal Serial Bus
UWB	Ultra-Wideband
WDF	Windows Driver Framework
WDM	Windows driver Model
WiMCA	WiMedia Convergence Architecture
WLP	WiMedia Logical Link Control Protocol

Table 1: Terms and Abbreviations

1 Executive summary

EUWB WP7 will provide Open Platforms for the ECMA 368/369 specification based on close to commercial chipsets. The platforms should be compliant with the ECMA 368/369 standard and should fulfil the European and world-wide regulatory requirements in their final version.

This deliverable has considered the requirements of the different application and research areas that will use the HDR Open Platform in the corresponding work packages within EUWB. In response, WP7 offers this platform specification with the intention that it might meet a broad range of the requirements and scenarios described by the corresponding work packages.

This document provides the Open Platform specification for the HDR (high data rate) communication platform deliverable for WP7. As such, it provides details of the HDR ‘Open Platform’, and provides a comprehensive description of the platform, including the physical layer, MAC layer, and MAC layer interface to the application layers.

2 Introduction

This deliverable describes the HDR “Open Platform” to be developed in EUWB by partners Wisair and TES Electronic Solutions. The development of the platform will follow a phased approach. Wisair and TES believe that such an approach will offer assurance of successful delivery of the platform.

The HDR platform will follow a modular approach with a split at the MAC / PHY interface. The MAC shall be implemented on a commercial ‘off the shelf’ FPGA development platform, whilst the PHY module (including both RF and Baseband functionality) will be provided as a daughter card. The MAC layer shall offer partners the ability to interface higher protocol layers above the MAC layer interface, however an IP based interface shall be provided to allow IP based application scenarios to be developed directly with the development platform interface.

The platform with the PHY integrated as a daughter card and connected at the MAC / PHY interface with the Xilinx ML507 FPGA development platform is shown in Figure 1. Access to the MAC layer by higher protocol layers on this platform is provided through the Gb Ethernet interface.



Figure 1: Modular Approach to HDR “Open Platform” (Prototype)

The HDR “Open Platform” specification is provided by section 3. It describes the implementation phases of the PHY module and the corresponding features supported by each of these phases, a detailed description of the PHY Module RF capabilities, and a detailed description of the MAC layer and the associated MAC layer interface and API offered by the platform.

3 HDR “Open Platform” Specification

This chapter will provide an overall description of the EUWB HDR “Open Platform” for readers interested in integrating it within a project solution for an application scenario verification system. Therefore the description that follows focuses on the characteristics and features of platform that are relevant to this activity.

3.1 High-level Structure

The overall structure of the HDR “Open Platform” architecture is shown on Figure 2. The architecture is defined as a combination of protocol layers, of which the lower layers are part of the “Open Platform” and higher layers may be integrated with them to create several systems:

- Application Interface (API)
- Host NDIS / Network Driver
- Medium Access Control layer (MAC),
- Physical layer (PHY)

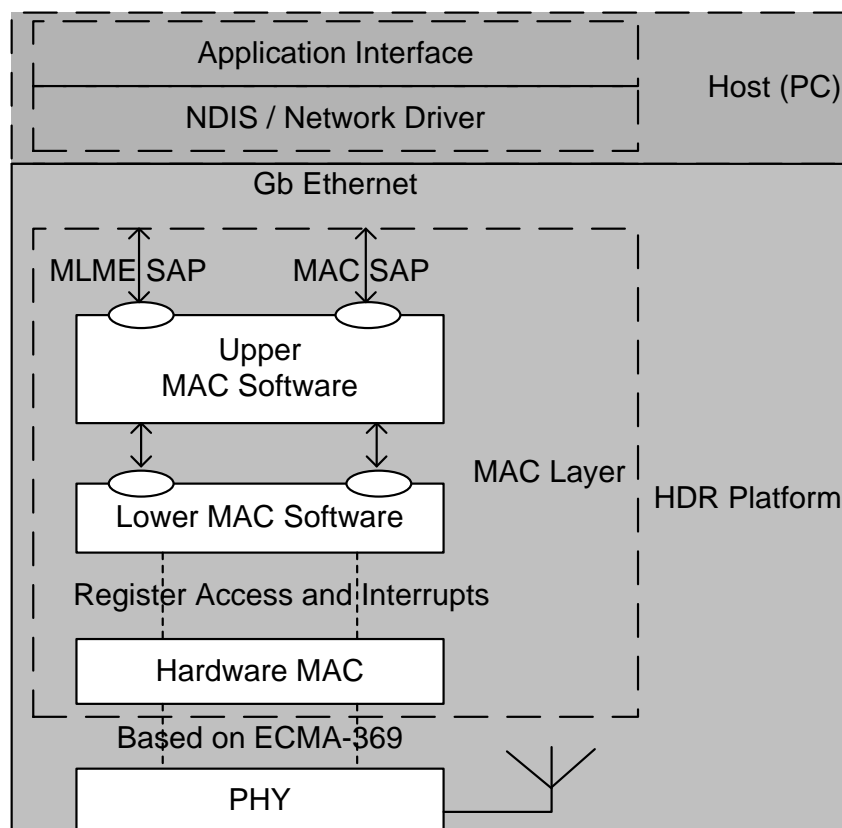


Figure 2: Overall layer structure of the HDR “Open Platform”

3.2 Implementation phases plan

The EUWB HDR “Open Platform” implementation is planned to be delivered in several phases, each with a different PHY HW, this is intended to ease integration and enable the introduction of additional desired features to the PHY HW. Originally, three (3) such phases were planned as shown in Table 2 below.

Feature	Phase 1 - Preliminary	Phase 2 – first usable platform	Phase 3 & 4 (and later phases)
Supported Frequencies - RF band Coverage	BG 1 (<5GHz)	BG 1 (<5GHz)	BG1 and at least BG 4, exact – TBD 0
Time frequency Codes	1-7	1-7	All (1-10)
Data Rates	All (53.3 – 480)	All (53.3 – 480)	All (53.3 – 480)
Short Preamble support	Available	Available	Available
Ranging and Location Awareness	None	None	Added to Phase 3 or 4 at level TBD 0

Table 2: PHY HW Features

3.3 Integration plan

Since the release of interim deliverable IR7.1.2, it has been necessary to revise the original plan and consolidate the number of integration phases. This decision was taken due to the complex RF calibration requirements of the Wisair 601 chipset. Since support for the Wisair 601 PHY does not offer any additional feature support to that of the Wisair 532 PHY, the revised integration plan as shown in Table 3 was proposed and agreed.

Stage	Original Plan ¹	Actual	Revised Plan	Comments
Initial integration (532 chip)	3Q-4Q 2008	1Q-2Q 2009 (supports single link only)	1Q-2Q 2009 (No change!)	Only internal to WIS-TES
Second integration (601 chip)	2Q 2009	4Q 2009 to 1Q 2010	Skip this step!	Complex RF calibration – not simple to integrate with FPGA ‘Open Platform’
Third integration (612 chip)	1Q-2Q 2010	2Q-3Q 2010 (Too late!)	² Integration starts Q1-Q2 2010	Supports simplified RF calibration Supports Ranging (phase 4)

Table 3: Revised Integration Plan

¹Dates are for EUWB integration, and not chips availability

²Expect to provide partners with a phased approach for support of ranging and DAA

3.4 General description

3.4.1 PHY General Description

The (V)HDR Open Platform is considered as a platform that enables short range, low power, high-speed wireless connectivity of up to 480 MBit/s by using the ultra wideband spectrum from 3.1 to 10.6 GHz. Due to different modulation schemes and spreading methods the platform supports data rates of 53.3 MBit/s, 80 MBit/s, 106.7 MBit/s, 160 MBit/s, 200 MBit/s, 320 MBit/s, 400 MBit/s and 480 MBit/s.

The UWB bandwidth from 3.1 to 10.6 GHz is divided into 14 bands, each with a bandwidth of 528 MHz greater than the minimum bandwidth 500 MHz of the UWB signal, so that the definition of UWB is satisfied. These 14 bands are grouped into 5 band groups with 3 bands in each band group excepting the last group which has only two bands. The sixth group is also defined within the spectrum of group three and four, consistent with usage within world wide spectrum regulations. The frequency coverage of the platform deployed in this project is described in chapter 3.5.

The MultiBand Orthogonal frequency Division Modulation (MB-OFDM) scheme is specified in the ECMA standard in order to transmit information with UWB. Each band consists of 128 sub-carriers with 4.125 MHz intervals: 100 data sub-carriers, 10 guard sub-carriers, 12 pilot sub-carriers for coherent detection and 6 null sub-carriers.

Two modulation schemes are employed for the data sub-carriers: QPSK (quadrature phase-shift keying) and DCM (dual carrier modulation). The quality of the UWB link and the transmission data rate depends on the deployed modulation scheme. For a high quality link (for data rate 320 MBit/s or higher) the DCM scheme, which is more robust against multipath fading channels, is preferred in order to maximize the data rate, while the lower quality link (data rate 200 Mb/s or lower) employs the QPSK scheme using lower code rates. Nevertheless, in an AWGN scenario, both modulation schemes behave identical. In the QPSK mode two bits are mapped to each sub-carrier while in the DCM mode four bits are mapped to two sub-carriers.

In order to improve the BER (bit error rate) two types of redundancy are used: Frequency-domain spreading (transmitting the same data on two separate sub-carriers within the same OFDM-symbol) and time-domain spreading (transmitting the same data over two bursts). Together with forward error correction (FEC) convolutional codes with code rates of 1/3, 1/2, 5/8 and 3/4 the transmission data rates can be adapted.

The coded OFDM symbols is then spread by hopping across the bands in each band group according to a hopping sequence which is predefined and called TFC (time frequency code). It should be noted that this hopping can be executed only within one band group. A hopping among several band groups is not possible. 10 TFCs are supported for each band group. Three types of TFCs are specified in ECMA standard: a) the OFDM symbols are hopped across all the three bands within one band group, referred to as Time-Frequency Interleaving (TFI); b) The OFDM symbols are interleaved over two bands within one band group, referred to as two-band TFI or TFI2; and c) only one single band of a band group is utilized during transmission, referred to as Fixed Frequency Interleaving (FFI).

In the case of data rate 53.3 Mb/s the QPSK modulation as well as FDS, TDS and FEC with a code rate of 1/3 are employed. The following calculation indicates how to achieve this data rate: deploying

QPSK modulation, 200 bits are mapped on 100 data symbols forming the payload part of one OFDM symbol. The FDS and TDS reduce the throughput in half respectively and 50 bits/symbol are obtained. The FEC works on six-symbol intervals, therefore six symbols times 50 bits/symbol result in 300 raw bits. Considering the FEC code rate 1/3 only 100 coded bits will be transmitted in six symbol intervals, i.e. 1.875 μ s. Thus, the throughput equals 53.3 Mb/s. To achieve a data rate of 480 Mb/s DCM is employed together with no FDS and no TDS. The FEC coding rate is 3/4. The calculation is analogue: The DCM mapping results in 200 bits/symbol. Due to no FDS and TDS these bits will be input to FEC without redundancy. FEC works on six-symbol intervals, hence, six symbols times 200 bits/symbol equals 1200 raw bits. The FEC code rate 3/4 reduces the raw bits to 900 coded bits which will be transmitted in 1.875 μ s resulting in a data rate of 480 Mb/s.

3.4.2 PHY layer partitioning

It is well known that MAC is a sub-layer of Data Link Layer defined in the OSI basic reference model. The MAC sub-layer relies on the service provided by the PHY via the PHY service access point (PHY SAP). In order to better describe the services and functions of PHY the PHY is also divided into two sub-layers defined in ECMA standard: a PHY convergence protocol (PLCP) sub-layer and a physical medium dependent (PMD) sub-layer. Figure 3 shows the Data Link Layer, PHY and their sub-layers.

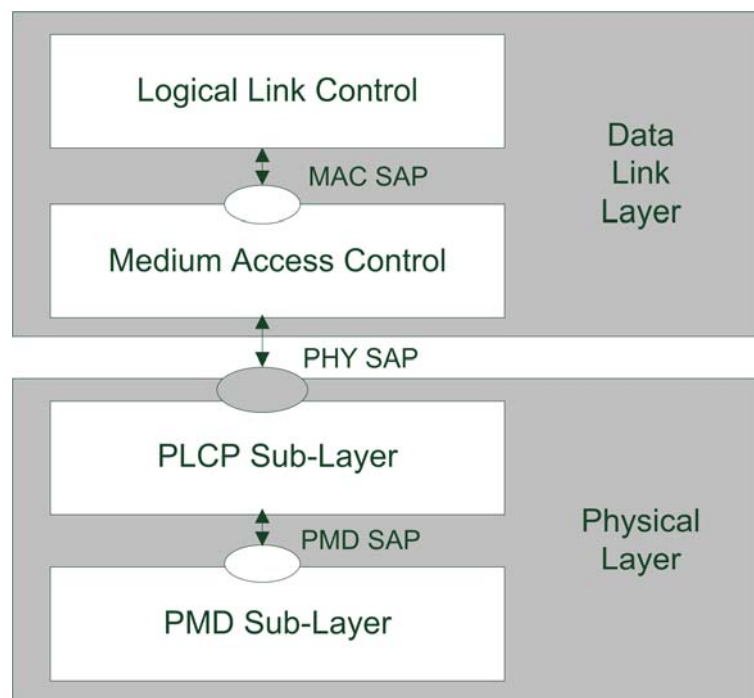


Figure 3: Physical layer partitioning

- PLCP sub-layer:

The PLCP sub-layer is an upper sub-layer in the PHY. It communicates with MAC and PMD sub-layer via the PHY SAP and PMD SAP, respectively. This sub-layer is defined so that the MAC can operate with minimum dependence on the PMD sub-layer. The physical layer convergence protocol (PLCP) of this sub-layer defines how to convert the PLCP service data units (PSDU) into a framing format suitable for sending and receiving user data, the PLCP protocol data unit (PPDU) and how to control the information among stations using the associated PMD device.

- PMD sub-layer:

The PMD sub-layer provides the actual transmission and reception of PHY via wireless medium between two or more stations. It provides also the modulation and demodulation of the transmission frame.

3.4.3 PHY frame structure

In terms of the OSI basic reference model the transmission data passes through each layer top-down with the header and/or tailer appended at each layer, and bottom-up by removing the header and/or tailer. The so called service data unit (SDU) is a data unit of the payload which a particular layer provides to the layer above. A protocol data unit (PDU) is defined as the transmission data at a layer with the overhead. As mentioned before, the PLCP defines a method to convert the PSDU to PPDU. This clause describes the PHY frame structure, i.e. the structure of the PPDU.

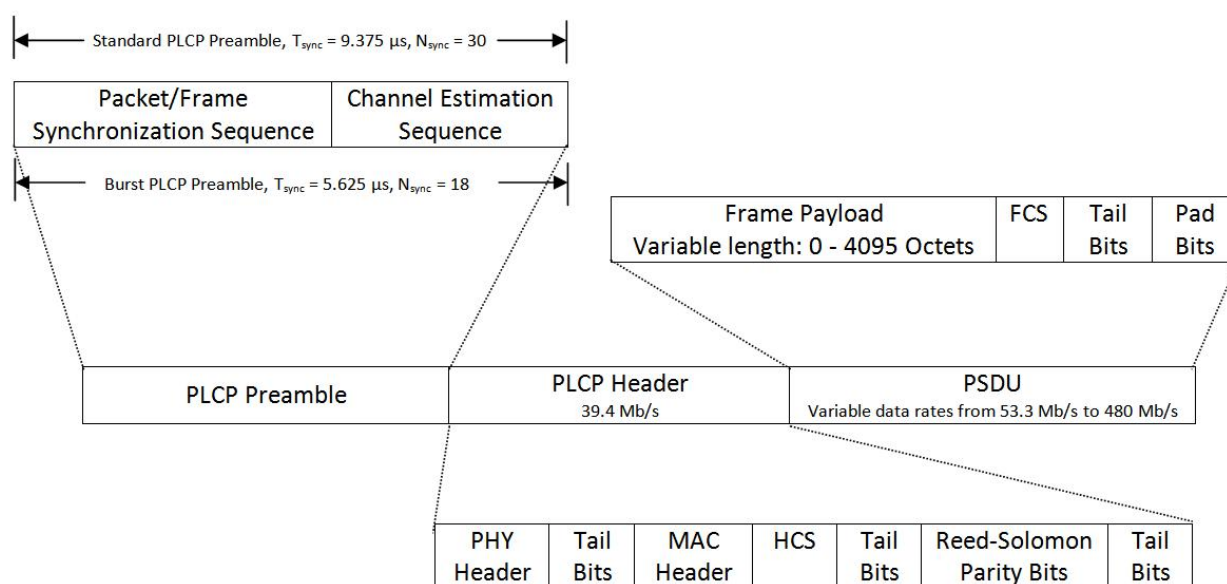


Figure 4: Standard PPDU structure

Figure 4 shows the structure of PPDU. As presented in this Figure, the PPDU is composed of three components: PLCP preamble, PLCP header and PSDU. The PLCP preamble and the PLCP header are appended ahead of the PSDU for the sake of providing a stable transmission. During the transmission the PLCP preamble is sent first, afterwards the PLCP header and the PSDU is the last to be sent. The PLCP header is transmitted always at data rate 39.4 Mb/s. The transmission data rate of PSDU is variable from 53.3 Mb/s up to 480 Mb/s, as presented before. Hence, the required data rate does not refer to the real physical transmission data rate of PPDU in the PMD sub-layer, but the data rate of PSDU. Of an octet the least-significant bit (LSB) is sent first.

- PLCP preamble

As the first transmitted component of PPDU the PLCP preamble can help the receiver in timing synchronization, carrier-offset recovery, and channel estimation. There are two types of PLCP preamble: standard PLCP preamble and burst PLCP preamble. The standard PLCP preamble can be used in both standard mode and burst mode while the burst preamble can be used only when a burst of packets is transmitted. For data rate 200 Mb/s or lower only the standard preamble may be used, even

if the burst of packets is transmitted. For higher data rate the first packet in the burst must use the standard PLCP preamble and the other packets may use either standard or burst PLCP preamble. The number of symbols and the duration of standard PLCP preamble are 30 and 9.375 μ s, respectively while those of the burst PLCP preamble are 18 and 5.625 μ s, as presented in Figure 4. Both types of preambles are composed of Packet/Frame synchronization sequence which is a time domain portion and channel estimation sequence referred to a frequency domain portion. The Packet/Frame synchronization sequence can be applied for the sake of packet acquisition and detection, coarse carrier frequency estimation, coarse symbol timing and synchronization within the preamble. While the channel estimation sequence can be used for estimation of the channel frequency response, fine carrier frequency estimation and fine symbol timing. The first sample of the first channel estimation symbol shall be used as timing reference point for range measurements. The principle of constructing the Packet/Frame synchronization sequence is similar with that of the channel estimation sequence; both are defined in ECMA 368 Standard.

- PLCP header

The PLCP header follows the PLCP preamble, is the second component to be transmitted in the PPDU. The PLCP header conveys the information about the PHY and the MAC which is necessary at the receiver in order to successfully decode the PSDU. The scrambled and Reed-Solomon encoded PLCP header is defined in Figure 5:

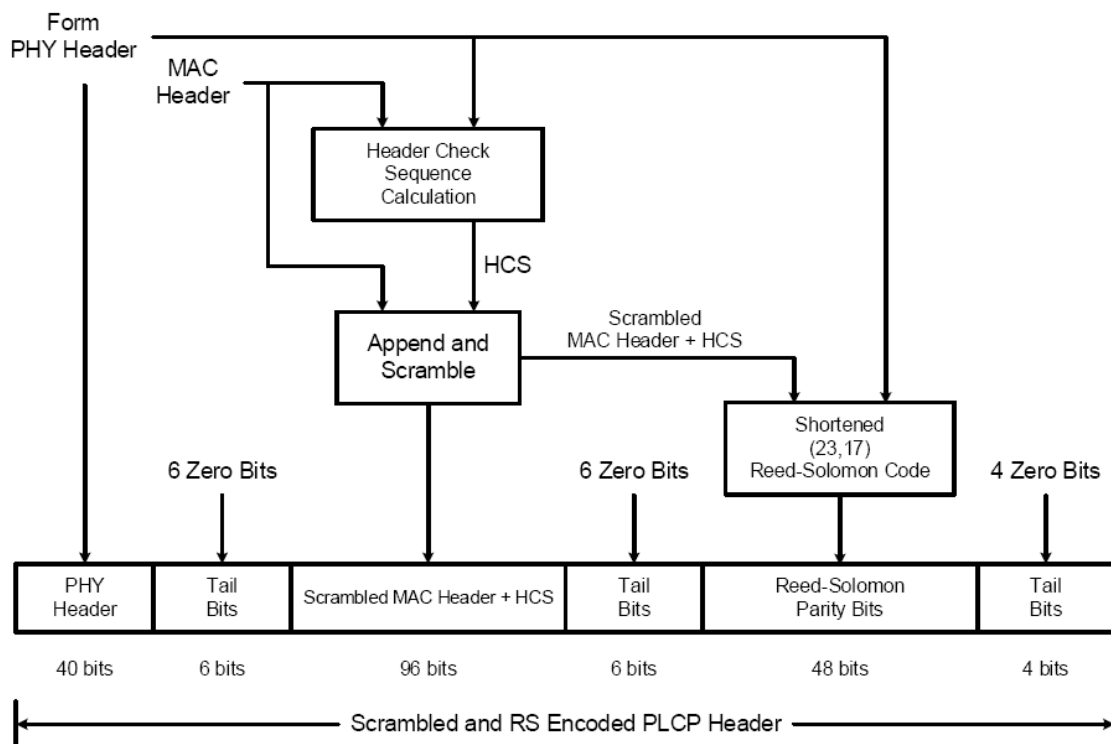


Figure 5: Block diagram of PLCP header construction, from [4]

- The first part of the PLCP header is the PHY header which is formatted based on information provided by the MAC. It contains information about the data rate of the MAC frame body, the length of the frame payload (which does not include the FCS), the seed identifier for the data scrambler, and information about the next packet- whether it is being sent in burst mode and whether it employs a burst preamble or not [4]. After the 40 bits PHY header 6 zero tail bits will be appended.

- b) A header check sequence (HCS) value is then calculated over the combined PHY and MAC headers. The resulting HCS value is appended to the MAC header. They are together scrambled, composed of 96 bits and follow the 6 zero tail bits, as the second major part of the PLCP header. Afterwards 6 zero tail bits are appended again.
- c) A shortened (23, 17) Reed-Solomon code is applied for the concatenation of PHY header, scrambled MAC header and HCS. Totally 6 octets are reserved for this part and they are as Reed-Solomon parity bits appended to the second 6 zero tail bits. At last 4 zero tail bits are appended again.

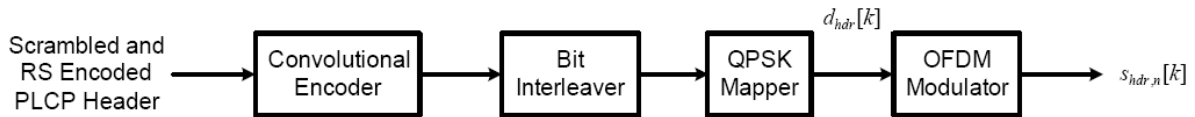


Figure 6: Encoding process for the scrambled, Reed-Solomon encoded PLCP header, from [4]

As shown in Figure 6, the resulting scrambled and Reed-Solomon encoded PLCP header is then encoded by a convolutional encoder ($R = 1/3, k = 7$), interleaved using a bit interleaver, passed through a QPSK mapper, and the resulting complex values are mapped onto the OFDM sub-carriers so that the baseband signal could be created.

- PSDU

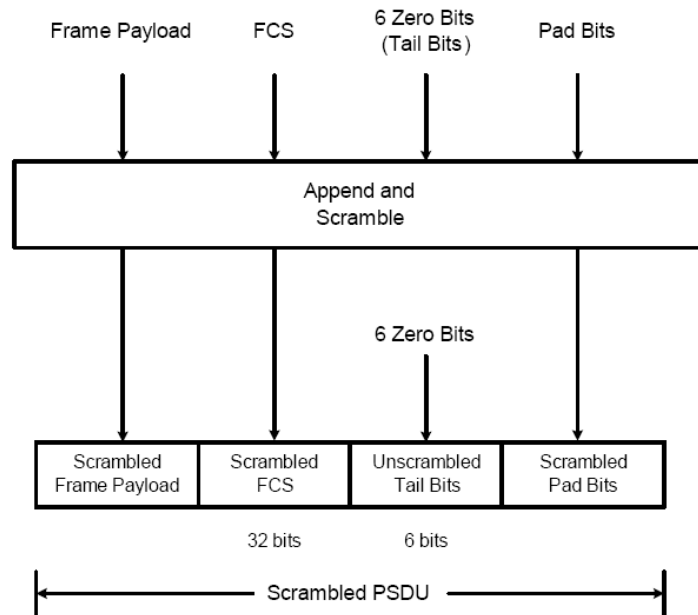


Figure 7: Block diagram of PSDU construction, from [4]

As the third and major component of PPDU the PSDU is composed of scrambled frame payload, scrambled frame check sequence (FCS), unscrambled tail bits and scrambled pad bits. The construction of PSDU is shown in Figure 7. The concatenation of frame payload, FCS, 6 zero tail bits and pad bits is scrambled according to a method described in [4]. Then the six scrambled tail bits shall be replaced by the six non-scrambled tail bits so that the PSDU is formed.

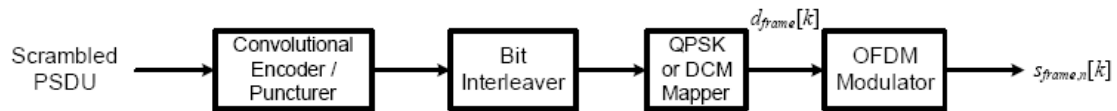


Figure 8: Block diagram of the encoding process for the scrambled PSDU, from [4]

As shown in Figure 8, the resulting scrambled PSDU is encoded by a convolutional code ($R = 1/3$, $k = 7$) and punctured to achieve the desired coding rate, interleaved by a bit interleaver, passed through either QPSK or DCM mapper depending on which data rate is desired, and finally, the resulting complex values are mapped onto the data sub-carriers of the OFDM symbol in order to create the real or complex baseband signal.

3.4.4 MAC General Description

The MAC is the lower sub-layer of the data link layer defined in OSI basic reference model. The MAC service is provided by the MAC service access point (MAC SAP) to a single MAC service client, usually a higher layer protocol or adaptation layer. The PHY provides service for the MAC by the PHY service access point (PHY SAP). The MAC general description is well presented in [4]. In this deliverable the focus locates on the relationship between MAC and PHY. The MAC sub-layer requires the following features provided by the PHY [4]:

- Frame transmission in both single frame and burst mode
- Frame reception for both single frame and burst mode transmission
- PLCP header error indication for both PHY and MAC header structures
- Clear channel assessment for estimation of medium activity
- Range measurement timestamps if MAC range measurement is supported.

The exchange of parameters between the MAC sub-layer and the PHY makes the frame transmission and reception possible. In single frame transmission the MAC sub-layer controls the frame timing fully, while in burst mode transmission the MAC sub-layer has only the control of the first frame timing and the PHY provides accurate timing for the remaining frames in the burst.

3.5 Description of PHY Module

The HDR “Open Platform” PHY is well defined by the ECMA 368 [4] and WiMedia Standards [3]. The following sub sections describe the PHY characteristics of most interest to implementation of the higher layers. It goes on to describe some additional features which are not fully standardised concerning location estimation.

3.5.1 Main PHY & RF Characteristics

3.5.1.1 Supported Frequencies:

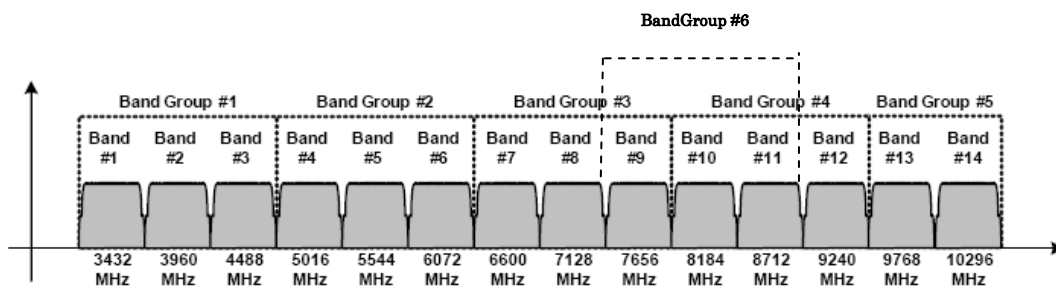


Figure 9: WiMedia Frequency bands plan

The WiMedia frequency plan is constructed of five (5) orthogonal Band-Group (BG) definitions and one additional overlapping band group (#6). Most Band-Groups are composed of three (3) separate bands except for BG #5 which has only two (2) bands. Figure 9 and Table 4 give explicit description of the WiMedia Frequency plan including Band-Width and center frequencies.

Not all bands are allowed for usage by the World Wide regulatory bodies. Particular attention has been given to BG #1 which may only be utilized in Europe for a limited time, except for utilizing Band #3 with Detect and Avoid (DAA). Further, most manufacturers and regulators seem to agree not to utilize BG #2 due to high density of wireless services in that spectral band.

The EUWB HDR “Open Platform” PHY & RF implementation in the first two phases will only support BG #1, below 5GHz. The next step PHY & RF implementation is planned to support both BG #1 and some BGs above 6GHz. The exact band covered in those next phases is TBD 0.

Band Group	BAND_ID	Lower Frequency (MHz)	Center Frequency (MHz)	Upper Frequency (MHz)
1	1	3168	3432	3696
	2	3696	3960	4224
	3	4224	4488	4752
2	4	4752	5016	5280
	5	5280	5544	5808
	6	5808	6072	6336
3	7	6336	6600	6864
	8	6864	7128	7392
	9	7392	7656	7920
4	10	7920	8184	8448
	11	8448	8712	8976
	12	8976	9240	9504
5	13	9504	9768	10032
	14	10032	10296	10560
6	9	7392	7656	7920
	10	7920	8184	8448
	11	8448	8712	8976

Table 4: WiMedia Band Plan

3.5.1.2 Time frequency Codes (TFC):

Several TFCs are defined by the WiMedia standard, as described in Table 5.

TFC	Num of Bands used	Max relative Power	Orthogonal in group	Comments
1,2	3	+5dB	No	
3,4	3	+5dB	No	Not recommended for usage
5,6,7	1	0dB	Yes	Also called FFC (Fixed Frequency Code)
8,9,10	2	+3dB	No	Each is orthogonal to one FFC.

Table 5: WiMedia TFC

Note, Table 5 only describes the TFCs for BG #1-#4 and #6. Further some TFCs were defined later on (TFC 8, 9, 10) while others were defined previously. Furthermore, some TFCs (3, 4) do not provide an additional technical advantage.

The EUWB HDR “Open Platform” PHY & RF implementation will support TFCs 1-7 in the first two phases and all TFCs in the later phases.

3.5.1.3 Data Rates: Coding & Modulation:

The WiMedia specification defines the data rates in Table 6. All of these rates are supported in all the platform implementation phases.

Data Rate (Mb/s)	Modulation	Coding Rate (R)	FDS	TDS	Info Bits / 6 OFDM Symbols N_{IBP6S}
53.3	QPSK	1/3	YES	YES	100
80	QPSK	1/2	YES	YES	150
106.7	QPSK	1/3	NO	YES	200
160	QPSK	1/2	NO	YES	300
200	QPSK	5/8	NO	YES	375
320	DCM	1/2	NO	NO	600
400	DCM	5/8	NO	NO	750
480	DCM	3/4	NO	NO	900

Table 6: WiMedia Data Rates

3.5.1.4 OFDM symbol mapping:

The WiMedia specification defines the OFDM symbol mapping based on Table 7. All of these requirements are supported in all the implementation phases. This section is informative only. These parameters are not available for change.

Parameter	Description	Value
fs	Sampling frequency	528 MHz
NFFT	Total number of sub-carriers (FFT size)	128
ND	Number of data sub-carriers	100
NP	Number of pilot sub-carriers	12
NG	Number of guard sub-carriers	10
NT	Total number of sub-carriers used	122 (= ND + NP + NG)
Δf	Sub-carrier frequency spacing	4.125 MHz (= fs/NFFT)
TFFT	IFFT and FFT period	242.42 ns (Δf^{-1})
NZPS	Number of samples in zero-padded suffix	37
TZPS	Zero-padded suffix duration in time	70.08 ns (= NZPS/fs)
TSYM	Symbol interval	312.5 ns (= TFFT + TZPS)
FSYM	Symbol rate	3.2 MHz (= TSYM ⁻¹)
NSYM	Total number of samples per symbol	165 (= NFFT + NZPS)

Table 7: OFDM symbol mapping

3.5.1.5 Frame structure:

The WiMedia specification defines the Frame structure as defined in Table 8. Two preambles are defined: a standard preamble and a burst preamble. The burst preamble shall only be used in burst mode when a burst of packets is transmitted, which are separated by a minimum inter-frame separation time;

- For data rates of 200 Mb/s and lower, all the packets in the burst shall use the standard preamble.
- However, for data rates higher than 200 Mb/s, the first packet shall use the standard preamble, while the remaining packets may use either the standard preamble or the burst preamble

3.5.1.6 Frame-related parameters

Parameter	Description	Value
N_{pf}	Number of symbols in the packet/frame synchronization sequence	Standard Preamble: 24, Burst Preamble: 12
T_{pf}	Duration of the packet/frame synchronization sequence	Standard Preamble: 7.5 μ s Burst Preamble: 3.75 μ s
N_{ce}	Number of symbols in the channel estimation sequence	6
T_{ce}	Duration of the channel estimation sequence	1.875 μ s
N_{sync}	Number of symbols in the Preamble	Standard Preamble: 30, Burst Preamble: 18
T_{sync}	Duration of the Preamble	Standard Preamble: 9.375 μ s Burst Preamble: 5.625 μ s
N_{hdr}	Number of symbols in the PLCP Header	12
T_{hdr}	Duration of the PLCP Header	3.75 μ s
N_{frame}	Number of symbols in the PSDU	$6 \times \left\lceil \frac{8 \times \text{Information_Length} + 38}{N_{IBP6S}} \right\rceil$
T_{frame}	Duration for the PSDU	$6 \times \left\lceil \frac{8 \times \text{Information_Length} + 38}{N_{IBP6S}} \right\rceil \times T_{SYM}$
N_{packet}	Total number of symbols in the packet	$N_{sync} + N_{hdr} + N_{frame}$
T_{packet}	Duration of the packet	$(N_{sync} + N_{hdr} + N_{frame}) \times T_{SYM}$

Table 8: OFDM Frame parameters

Both Preamble types are supported in all implementation phases. Usage of the short preamble in packet bursts of data rate above 200Mbps is controlled by the MAC.

3.5.1.7 Transmit Power Control:

The WiMedia specification defines the Transmit Power Control (TPC) as defined in Table 8.

<i>TXPWR_LEVEL</i>	<i>TX Power Attenuation for TFI Modes</i>	<i>TX Power Attenuation for TFI2 Modes</i>	<i>TX Power Attenuation for FFI Modes</i>
<i>0</i>	<i>0 dB</i>	<i>0 dB</i>	<i>0 dB</i>
<i>1</i>	<i>2 dB</i>	<i>2 dB</i>	<i>2 dB</i>
<i>2</i>	<i>4 dB</i>	<i>4 dB</i>	<i>4 dB</i>
<i>3</i>	<i>6 dB</i>	<i>6 dB</i>	<i>6 dB</i>
<i>4</i>	<i>8 dB</i>	<i>8 dB</i>	<i>8 dB</i>
<i>5</i>	<i>10 dB</i>	<i>10 dB</i>	<i>RESERVED</i>
<i>6</i>	<i>12 dB</i>	<i>RESERVED</i>	<i>RESERVED</i>
<i>7</i>	<i>RESERVED</i>	<i>RESERVED</i>	<i>RESERVED</i>

Table 9: Transmit Power Control

In either case, the relative accuracy of change in transmit power attenuation shall be the maximum of ± 1 dB or $\pm 20\%$ of the change in the attenuation (in the dB scale).

The TPC of the implementations supports the TPC requirements.

3.5.2 Additional PHY & RF features

This section describes additional optional features which are considered to be of interest in the EUWB project.

3.5.2.1 Ranging & Location Awareness

The WiMedia PHY specifications define an optional measurement for "RANGING AND LOCATION AWARENESS", based on propagation delay measurements that are intended, theoretically, to achieve an accuracy of ± 60 cm in line of sight (based on the signal BW, and relevant clock rate). The timing reference point is defined as the beginning of the first channel estimation symbol in the PLCP preamble.

Support for ranging measurement in the PHY is based counting the arrival time and storing this in a register. Such a counter may be clocked by a frequency source of 528 MHz to achieve a 56.8 cm ranging uncertainty. To provide increased precision, optional implementations may clock bit 2 at 1056 MHz (28.4 cm), bit 1 at 2112 MHz (14.2 cm), or clock bit 0 at 4224 MHz (7.1 cm).

The first and second PHY implementations will **not** support ranging or location awareness. The third or fourth implementation phase is intended to provide support for this feature, the implementation clock and access limitation will be TBD 2.

3.5.2.2 Detect And Avoid (DAA)

A Detect and Avoid feature is the ability to analyze the received signal, identify the reception of higher priority wireless services and change the transmission attributes accordingly (for example, change band, or remove relevant tones).

The exact procedure for DAA has not been defined in any standard.

The PHY implementation of the second phase supports spectral calculations for DAA decisions. Yet the implementation of this feature in the platform has yet to be decided upon - TBD 3.

3.5.3 PHY MAC Interface

As agreed in the EUWB project, the PHY MAC interface information shall be provided to TES only in a separate document.

3.6 Description of MAC Layer

The MAC core is implemented in a combination of software written in ‘C’, targeted at a PPC embedded processor, and digital hardware implemented in VHDL. The MAC core digital hardware has been developed by TES Electronic Solutions on a Virtex-5 XC5VFX70T-1CES FPGA development platform, and includes the ECMA 369 MAC PHY interface [5].

In the course of WP7, the baseband / PHY as specified in section 3.4 has been integrated with the MAC core to provide the HDR “Open Platform”. Partners will be provided with object code for the MAC core for execution on the target development platform.

3.6.1 Development Accessories

A number of development accessories are available from TES Electronic Solutions to support and test the MAC HDR core.

3.6.1.1 FPGA Development Platform

In order to minimise development effort and design risk, TES Electronic Solutions have developed the HDR UWB MAC core on an off-the-shelf development platform. The Xilinx ML507 FPGA Development Board is shown in Figure 10 and Figure 11.

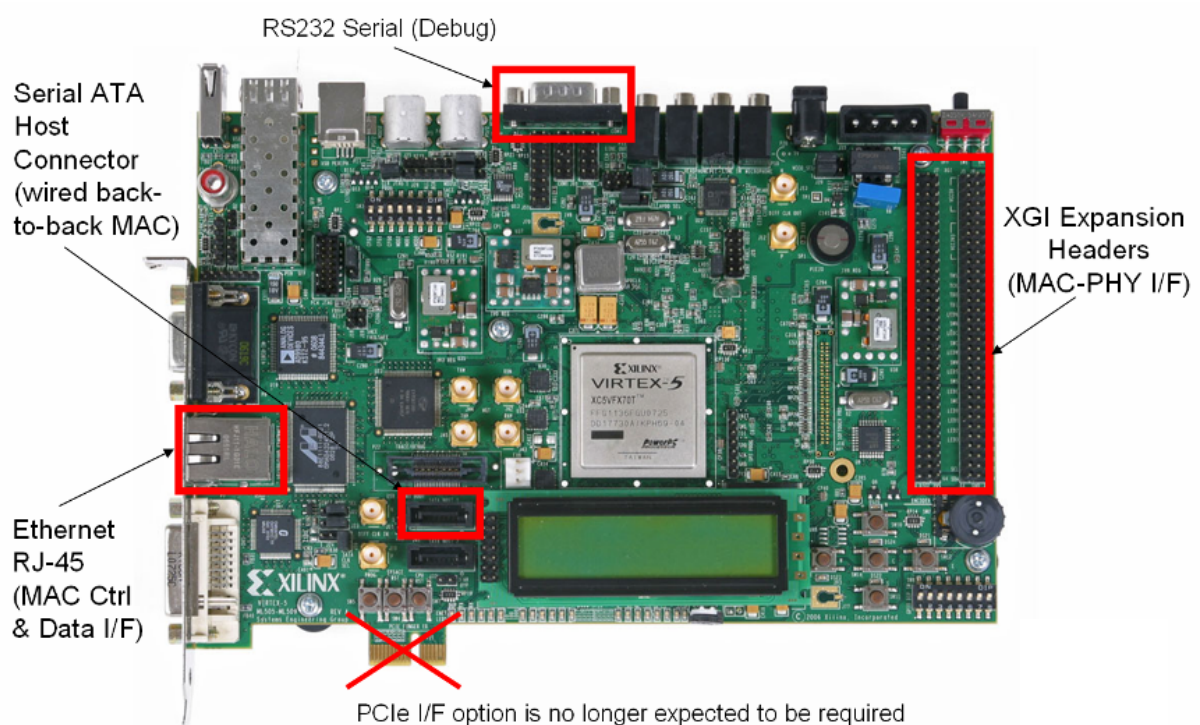


Figure 10: Xilinx ML507 FPGA Development Platform component side

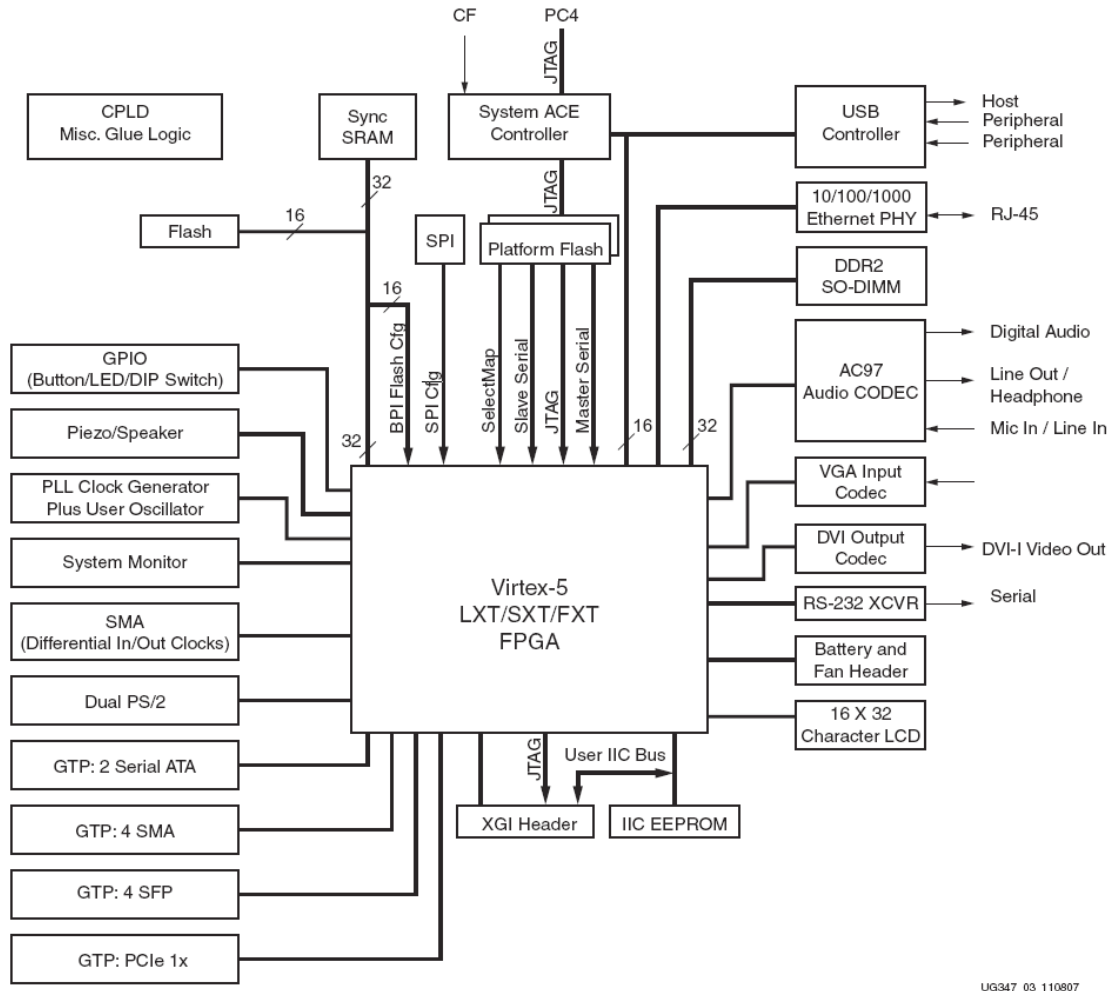


Figure 11: ML507 Board block diagram

3.6.1.2 TMKiiMicro Kernel Version 2

Many embedded systems are designed using UML, a modelling language and graphical notation, as specified by the OMG (Object Management Group). UML helps software engineers to specify, visualise, and document models of software systems, including their structure and design.

The MAC core software components have been designed and generated using *Rhapsody in C* UML 2.0 modelling and development tool, provided by Telelogic.

The 2nd Generation TES Micro-Kernel (TMKii) is designed to directly support UML-based communication protocol stack designs. It supports task-based event-driven state machines and is targeted at model driven development of embedded software implementations.

3.6.1.3 Chekmate2 Test Tool

Chekmate2 is a test harness used to perform regression testing on software built around TMKii. The harnessing of the protocol stack software is integrated with the automatic code generation phase of the *Rhapsody in C* tool. Chekmate2 test scripting uses the Perl language as its basis, to allow sophisticated pattern matching, looping and conditional execution.

The protocol stack harnessed with Chekmate2 is driven through test scripts written in Perl, and utilising a number of specialised library extensions. These extensions provide facilities to:

- Send and receive messages to and from multiple sources (e.g. different layers or instances of the stack),
- Check and validate received messages.
- Perform conditional execution based on variables, constants, etc.

Communication between the scripts and the harnessed instances of the target protocol stack(s) is via Posix compliant sockets. These are available under MS Windows or other Posix compliant OSs (such as Unix or Linux). The socket interface is used for testing PC-based simulations of the stack, and could also be used to test a stack simulation running on another system (such as a different PC, UNIX workstation or target hardware).

3.6.1.4 Port Patch Panel (P3), sockets broker tool

Port Patch Panel (P3) tool provides a method of connecting TCP/IP ports, serial ports and parallel ports together and transporting Chekmate2 Test Interface (CTI) messages over these connections.

P3 is used in conjunction with PC and embedded applications that communicate using the CTI protocol. Some obvious examples being Chekmate2, the PC stack simulations, PC MMI simulations and embedded development systems.

3.6.2 MAC Core Overview

The MAC core has been designed and implemented with an architectural focus on functional completeness and performance. The following sub-sections discuss the partitioning of the MAC between software and hardware, and describe the key features of each.

3.6.2.1 High-level MAC Architecture and Partitioning

The MAC is divided into three parts:

- The Hardware MAC which provides procedures to manage small-scale transactions on the medium
- The Lower MAC (LMAC), which manages multiple streams of frames to be sent to and received from the medium, together with immediate control of the HMAc
- The Upper MAC (UMAC), which manages protocol exchanges with peer MAC entities.

The UMAC exposes the MLME-SAP and the MAC-SAP. The functionality is specified in reference [4], with some exceptions.

The natural interface between the UMAC and the LMAC is internal, in the sense that they require access to common data structures. This is for efficient communication and to avoid duplication. Moreover, they share access to common helper functions, for example for access to hardware registers.

Figure 13 provides a simplified overview of the HMAC hardware subsystem.

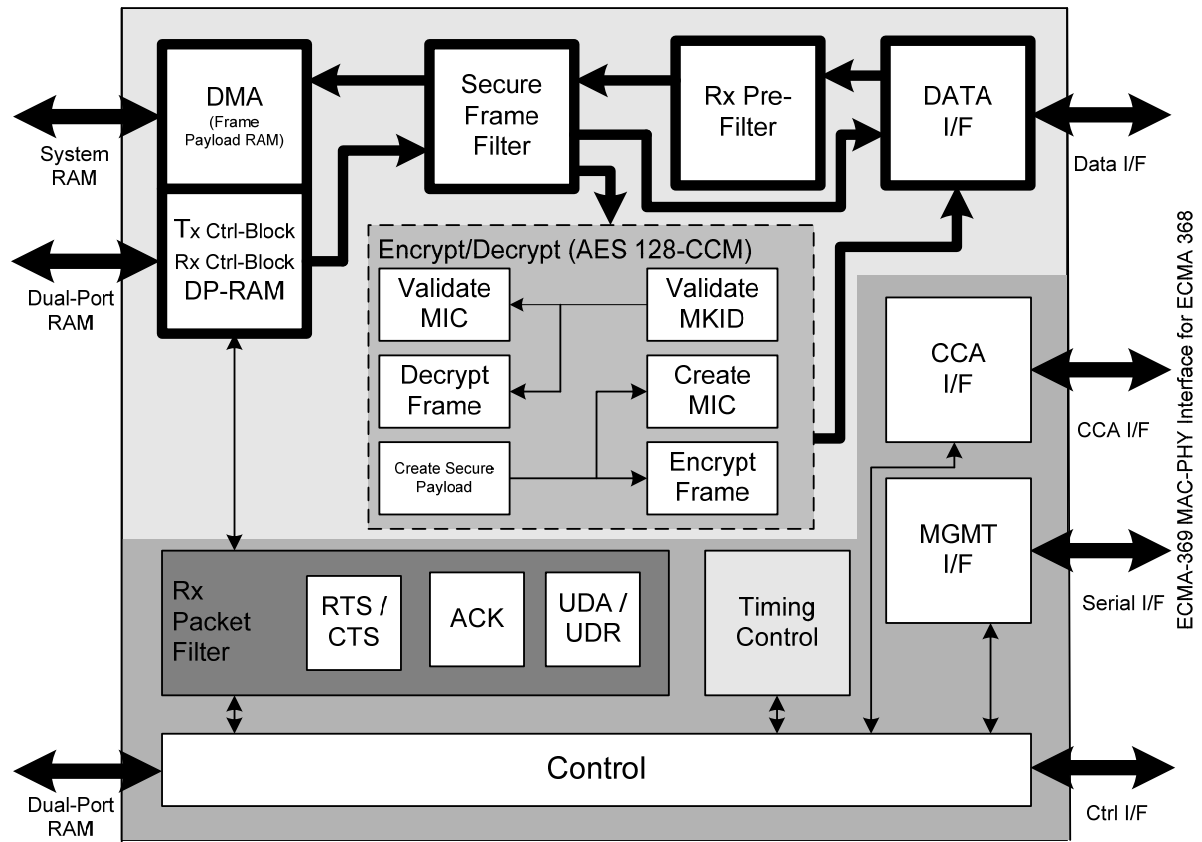


Figure 13: Overview of HMAC Core

The HMAC core performs real time processing of transmit and receive data and control interfaces between the software subsystem and the physical interface.

3.6.2.3 High-level Features

The key capabilities of the MAC core that will be available on the HDR open platform are;

Attribute	Supported (Y/N)
PCA	N ¹
Hard DRP	Y
Soft DRP	Y
Block Ack	N ¹
Explicit DRP Negotiation	Y
Hibernation Anchor	N
Probe	N ¹
Link Feedback	Y
Range Measurement	Y-TBD 1
Security levels M0	Y
Security levels M1	N ¹
Security levels M2	N ¹

Table 10 MAC Core High-level Features Supported by HDR Open Platform

3.6.3 MAC Core Interface Description

All messages (events) provided in the MAC core interface take the form of a variant to a service primitive, using the common '*req*', '*cfm*', '*ind*', '*rsp*' notation where;

A '*req*' is a request message received from a higher component layer.

A '*cfm*' is a confirmation message sent to a higher component layer as a result of processing a '*req*'. All '*req*' messages have a corresponding '*cfm*' message i.e. '*req*' and '*cfm*' are messages symmetrical.

An '*ind*' is an indication message sent to a higher component layer.

A '*rsp*' is a response message received from a higher component layer as a result of processing an '*ind*'.

Not all '*ind*' messages require a '*rsp*' message i.e. '*ind*' and '*rsp*' messages can be asymmetric.

¹ These features are available in the MAC core, but there are no plans to make this available on the open platform

3.6.3.1 Service Primitives for the Upper MAC

This section describes the messages that may be used at the service interface to the MAC Core by a client software component.

The service interface provided in this document has been derived from analysis and high-level design of the component layer based on the specification versions listed in the references section, found at the end of this document.

The primitives rely on three categories of datatypes:

- Types that apply specifically to the MAC core.
- Types that apply to the software protocol stack as a whole are global types.
- Base types such as bool (boolean) and u8 (unsigned 8-bit integer) are TMKii Kernel Types.

These types are shown in subsequent sections.

3.6.3.2 Events for the MAC core Interface

These Events correspond to the architecture described above. Separating the call from the response allows the MAC to provide concurrent management services for the clients.

3.6.3.2.1 MLME_Reset_req

This primitive requests the reset of the UMAC sublayer data path and its management entity and will result in the reset of the LMAC sublayer and the MIB in the HMAC interface.

The UMAC sublayer resets both the transmit and receive state machines, the management entity and MIB to the power on states with default values using the LMAC interface. The UMAC sublayer discards all MSDUs and their fragments, if any, that are buffered for transmission to a peer MAC sublayer or delivery to the UMAC client.

Until the MLME receives other primitives, the UMAC sublayer will not perform any transmit or receive operations. The MLME issues an MLME_Reset_cfm when the reset has completed, to reflect the results of the reset request.

Message Parameters

This message has no parameters.

3.6.3.2.2 MLME_Reset_cfm

This primitive reports the results of a reset procedure. This primitive is generated by the MLME as a result of an MLME_Reset_req at the completion of the reset operation.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	Indicates the result of the corresponding MLME_Reset_req. Valid result codes are: MAC_SUCCESS, MAC_FAILURE.

3.6.3.2.3 MLME_Scan_req

This primitive begins a scan operation.

Message Parameters

Name	Type	Description
u8PhyChannel	u8	The physical channel to be scanned.
ScanState	ScanState_e	Sets the scan state in the MLME-SCAN primitive.

3.6.3.2.4 MLME_Scan_cfm

This primitive confirms that a scanning state change initiated by the MLME_Scan request has been successfully completed or that the state change attempt has failed.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	Result of the service request.

3.6.3.2.5 MLME_ScanPLCPheaderRcvd_ind

This indication informs the MAC Client that a PLCP header was received. The indication is only received when the scan state is not SCAN_DISABLED.

Message Parameters

Name	Type	Description
psSoftDescriptorChain	SoftDescriptorChain_s*	This contains the PLCP header.
u8PhyChan	u8	The physical channel in use.
u16BPSTOffset	u16	The offset of the start of a received frame relative to the BPST of the device, measured in microseconds.
u8LQI	u8	The Link Quality Indication.
u8RSSI	u8	The Receive Signal Strength Indication.

3.6.3.2.6 MLME_SuperframeTick_ind

The MAC layer propagates a superframe timer tick, which is initially generated from the Hardware MAC as an interrupt, and sends it as an indication up to the upper layers.

Message Parameters

This message has no parameters.

3.6.3.2.7 MLME_BeaconStart_req

This primitive instructs the MAC sublayer to begin beacon transmission on the specified channel.

Message Parameters

Name	Type	Description
u8Phychan	u8	The physical channel on which the beacon will be transmitted or received.

3.6.3.2.8 MLME_BeaconStart_cfm

This primitive confirms the completion of an MLME-BEACON-START request. This confirmation indicates whether the first beacon has been transmitted.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	eResultCode indicates whether beacons have begun to be successfully transmitted. If the start beacon transmission is not successful, the operation returns MAC_FAILURE or MAC_FAILURE_NO_SLOTS, else the operation returns MAC_SUCCESS.

3.6.3.2.9 MLME_BeaconStop_req

This primitive causes the MAC sublayer to stop transmitting beacons. It has no parameters. The primitive is generated by the WiMCA (DME) to stop beacon transmission that was started with the MLME_BeaconStart_req.

Message Parameters

This message has no parameters.

3.6.3.2.10 MLME_BeaconStop_cfm

This primitive confirms that beacon transmission has been stopped by the UMAC after the WiMCA (DME) issued the MLME_BeaconStop_req. eResultCode indicates whether the beacon operation successfully stopped.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	ResultCode indicates whether the beacon operation successfully stopped. If ending beacon transmission is not successful, the service returns MAC_FAILURE, else it returns MAC_SUCCESS.

3.6.3.2.11 MLME_BeaconChangeChannel_req

This primitive causes the MAC sublayer to change the channel on which it transmits beacons and all other frames.

Message Parameters

Name	Type	Description
u8Phychan	u8	The physical channel on which the beacon will be transmitted or was received.
u8ChannelChangeIEcount	u8	The number of superframes in which to include a Channel Change IE prior to changing channels.

3.6.3.2.12 *MLME_BeaconChangeChannel_cfm*

This primitive confirms that the MLME has changed PHY channels as a result of a MLME-BEACON-CHANGE-CHANNEL request.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	Indicates the result of the corresponding MLME-BEACON-CHANGE-CHANNEL request.

3.6.3.2.13 *MLME_Beacon_ind*

This indication informs the MAC Client that a beacon was received or transmitted. The indication occurs any time the device receives a beacon or transmits its own beacon.

Message Parameters

Name	Type	Description
psSoftDescriptorChain	SoftDescriptorChain_s*	This holds the beacon frame.
u8PhyChan	u8	The physical channel in use.
u16BPSTOffset	u16	The offset of the start of a received beacon relative to the BPST of the device, measured in microseconds.
eBeaconType	BeaconType_e	The type of beacon reported, indicating whether a received beacon was classified as a neighbour or an alien, or if the beacon was transmitted by the device.
u8LQI	u8	Link Quality Indicator.
u8RSSI	u8	Received Signal Strength Indicator.
u16Length	u16	The length of the frame.

3.6.3.2.14 MLME_BeaconMergeBP_req

This primitive allows the DME to request initiation of the merging process of two previously separate BPs.

Message Parameters

Name	Type	Description
u16BPSToffset	u16	The offset of the start of a received beacon relative to the BPST of the device, measured in microseconds.

3.6.3.2.15 MLME_BeaconMergeBP_ind

This indication informs the DME that a neighbour has announced it will change its BPST as a result of a BP merge operation.

Message Parameters

Name	Type	Description
u16BPSToffset	u16	The offset of the start of a received beacon relative to the BPST of the device, measured in microseconds.
u8BPMoveCountDown	u8	The number of superframes before a BP merge commences, based on a neighbour's BP Switch IE.

3.6.3.2.16 MLME_BeaconMergeBP_cfm

This primitive confirms that a BP merge has completed or terminated.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	ResultCode indicates whether the merge operation successfully completed or was terminated.

3.6.3.2.17 MLME_BeaconPeriodExpired_ind

The MAC layer propagates this indication to signify the end of the current beacon period. The indication is initially generated from the LMAC by counting Hardware MAC MAS interrupts, and sends it as an indication up to the upper layers.

Message Parameters

This message has no parameters.

3.6.3.2.18 MLME_ASIEadd_req

This primitive changes the beacon content by adding an ASIE.

Message Parameters

Name	Type	Description
ASIEHandle	u16	A handle associated with an ASIE that has been added to the beacon content.
pau8ASIEdata	u8*	A variable size array containing ASIE data.
PositionAdvice	u16	The recommended position of the ASIE in the beacon.

3.6.3.2.19 MLME_ASIEadd_cfm

This primitive confirms the result of an operation to add an ASIE to the beacon in response to an MLME-ASIE-ADD request.

Message Parameters

Name	Type	Description
u16ASIEHandle	u16	A handle associated with an ASIE that has been added to the beacon content.
eResultCode	MlmeResultCode_e	Indicates the result of the corresponding request.

3.6.3.2.20 *MLME_ASIEremove_req*

This primitive instructs the MAC sublayer to remove an ASIE from the beacon content.

Message Parameters

Name	Type	Description
ASIEHandle	u16	A handle associated with an ASIE that has been added to the beacon content.

3.6.3.2.21 *MLME_ASIEremove_cfm*

This primitive confirms that the attempt to remove an ASIE from the beacon in response to an MLME-ASIE-REMOVE request has completed.

Message Parameters

Name	Type	Description
ASIEHandle	u16	A handle associated with an ASIE that has been added to the beacon content.
eResultCode	MlmeResultCode_e	Indicates the result of the corresponding request.

3.6.3.2.22 *MLME_ASIE_ind*

This primitive indicates the reception of an ASIE from a neighbour.

Message Parameters

Name	Type	Description
pau8ASIEData	u8*	A variable size array containing ASIE data.
Srcaddr_s	tEUI48	Holds the address of the device that sent the ASIE data.

3.6.3.2.23 *MLME_LinkEvent_req*

This primitive enables or disables the observation of link events for a specified link.

Message Parameters

Name	Type	Description
RemoteEUI	tEUI48	Specifies the EUI-48 of the remote device in the link.
MonitorState	u8	Specifies whether link event observation is active or inactive for the specified link.

3.6.3.2.24 MLME_LinkEvent_ind

This indication informs the DME that an event occurred for a link with monitoring enabled.

Message Parameters

Name	Type	Description
u8AccessMethod	u8	The access method used for transmission or receipt of a frame.
boBeacon	Bool	Indicates whether a received frame was a beacon.
u16BPSTOffset	u16	The offset of the start of a received frame relative to the BPST of the device, measured in microseconds.
u8LinkEventType	u8	The type of link event that occurred on a link being monitored.
u32PayloadSize	u32	Size of a transmitted or received frame payload.
u8PHYRate	u8	PHY data rate at which a frame was transmitted or received.
RemoteEUI	tEUI48	Specifies the EUI-48 of the remote device in the link.
u32RetryCount	u32	Retry count for this transmission.
u8ReceivedErrorInfo	u8	Provides additional information for a RECEIVE_ERROR.
u8DeliveryID	u8	The Delivery ID associated with a transmitted or received frame. See [1], s16.2.1.5.

3.6.3.2.25 *MLME_LinkEvent_cfm*

This confirmation indicates that a request to enable or disable link monitoring initiated by the Link Event request has been completed.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	Indicates the result of the corresponding request.

3.6.3.2.26 *MLME_DRP_req*

This primitive requests the creation of a new reservation or the modification or release of an existing reservation.

Message Parameters

Name	Type	Description
DestEUI	tEUI48	Identifies the respondent (either a single device or a multicast group) in the DRP negotiation initiated by the device identified by SrcEUI.
StreamIndex	u8	Identifies a stream from the device identified by SrcEUI to the device(s) identified by DestEUI.
ReservationType	u16	HARD, SOFT, PRIVATE, or PCA as specified by Table 106 of [1].
ReservationBitmapTable_st	Reservationbitmactable_s	Specifies the MASs desired or obtained for the reservation.
FinalReservation	Bool	Valid for multicast reservations, only. When TRUE, the final reservation is signaled to the multicast recipients.
Explicit	bool	Controls whether DRP negotiation is implicit or

explicit.

3.6.3.2.27 MLME_DRP_ind

This primitive requests the creation of a new reservation or the modification or release of an existing reservation.

Message Parameters

Name	Type	Description
DestEUI_st	tEUI48	Identifies the respondent (either a single device or a multicast group) in the DRP negotiation initiated by the device identified by SrcEUI.
SrcEUI_st	tEUI48	Identifies the DRP negotiation initiator.
u8StreamId	u8	Identifies a stream from the device identified by SrcEUI to the device(s) identified by DestEUI.
ReservationType_et	u16	HARD, SOFT, PRIVATE, or PCA as specified by Table 106 of [1].
ReservationBitMAP_st	Reservationbitmactable_s	Specifies the MASs desired or obtained for the reservation.
boExplicit	Bool	Controls whether DRP negotiation is implicit or explicit.

3.6.3.2.28 MLME_DRP_rsp

The DME uses this primitive to respond to a request for the creation of a new reservation or the modification or release of an existing reservation.

Message Parameters

Name	Type	Description
DestEUI_st	tEUI48	Identifies the respondent (either a single device or a multicast group) in the DRP negotiation initiated by the device identified by SrcEUI.

SrcEUI	tEUI48	Identifies the DRP negotiation initiator.
u8StreamIndex	u8	Identifies a stream from the device identified by SrcEUI to the device(s) identified by DestEUI.
ReservationType_et	u16	HARD, SOFT, PRIVATE, or PCA as specified by Table 106 of [1].
ReservationBitmapTable_st	Reservationbitmaptable_s	Specifies the MASs desired or obtained for the reservation.
boExplicit	Bool	Controls whether DRP negotiation is implicit or explicit.
Reasoncode_et	u16	Additional completion status information for the MLME request. ACCEPTED = 0, CONFLICT = 1, PENDING = 2, DENIED = 3
eResultCode	MlmeResultCode_e	Completion status of the corresponding request.

3.6.3.2.29 MLME_DRP_cfm

This function takes the result code from the MLME_DRP_cfm and generates a client message. The possible outcomes are either a cfm message replying to the client's reserve_req or a modify_ind otherwise. In any case, the data is two items, an eResultCode and the current state of the reservation, in terms of bandwidth and latency.

The client may have allowed some leeway between the desired and the minimum bandwidth. Also the actual latency may be better than the client's request. In any case, the client must be informed of the current state.

For FAILED, the result code is WCA_FAILURE.

For MODIFIED, the modified allocation is compared against the client's original request and the result is either WCA_SUCCESS or WCA_FAILURE. In this case, the (modified) reservation must be cancelled.

TIMEOUT is translated to WCA_FAILURE.

SUCCESS is translated to WCA_SUCCESS.

Message Parameters

Name	Type	Description
AvailabilityMap_st	Reservationbitmactable_s	Specifies the MASs available for DRP reservation.
eResultCode	MlmeResultCode_e	Completion status of the corresponding request.
Reason_et	u16	Additional completion status information for the MLME request..

3.6.3.2.30 MLME_DRPAvailabilityChange_ind

Sent by the MAC if the overall Reservation map has changed. It informs the client of the currently available MAS.

Message Parameters

Name	Type	Description
AvailabilityBitmap_st	Reservationbitmactable_s	Specifies the MASs available for DRP reservation.

3.6.3.2.31 MLME_MIBGet_req

This primitive requests information about a given attribute.

Message Parameters

Name	Type	Description
eMIBattribute	MAC_MIBType_e	Selects the MIB attribute that should be read.
puBuffer	MAC_MIB_pu	<<No Model Data>>
u32BufferLength	u32	<<No Model Data>>

3.6.3.2.32 MLME_MIBGet_cfm

This primitive reports the results of an information request about the MAC.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	Indicates the result of the MLME request.
MIBattribute	MAC_MIBType_e	Selects the MIB attribute that was read.

3.6.3.2.33 MLME_MIBSet_req

This primitive attempts to set the indicated MIB attribute to the given value.

Message Parameters

Name	Type	Description
eMIBattribute	MAC_MIBType_e	Selects the MIB attribute that should be set.
puBuffer	MAC_MIB_pu	The new value for the MIB element. Passed by value so the sender does not need to maintain static storage.

3.6.3.2.34 MLME_MIBSet_cfm

This primitive reports the results of an attempt to set the value of an attribute in the MIB.

Message Parameters

Name	Type	Description
eResultCode	MlmeResultCode_e	Result of the service request.
MIBattribute	MAC_MIBType_e	Indicates the MIB attribute that should was set.

3.6.3.2.35 MLME_RangeMeasurement_req

This primitive is used to initiate one or more consecutive ranging measurements. The client will use this primitive to initiate range measurement with a neighbouring device. Parameter u8RMN may be one (for a simple estimate) or greater than one, so that the results of repeated measurements can be used to improve accuracy. As a result of receiving this primitive, The MAC will generate frames to deliver over the medium to carry out the requested range measurements, and then delivers the results to the client as described in 3.6.3.2.36.

Message Parameters

Name	Type	Description
DestEUI	tEUI48	Identifies the EUI of the remote device for which to perform the range measurement.
u8RMN	u8	Number of range measurements requested to be performed.

3.6.3.2.36 MLME_RangeMeasurement_cfm

This primitive reports the results of a range measurement operation. The MAC sends this primitive to the client when it has completed a requested range measurement operation. The client may use a single measurement result to calculate a single range estimate. However, in order to resolve errors and ambiguities, the client may use averaging over several range measurements in order to obtain a more accurate estimate of range in the case of point-to-point applications. The range measurement accuracy may be further improved through the use of triangulation in the case of point-to-multi-point systems.

Message Parameters

Name	Type	Description
au8MeasurementResultSet	Array of u8	A variable size array of octets containing the Measurement Result Set as specified in 3.6.3.2.38.
eResultCode	MlmeResultCode_e	Indicates the result of the corresponding MLME-RANGE- MEASUREMENT-request. Valid result codes are ; MAC_SUCCESS, MAC_FAILURE

3.6.3.2.37 MLME_RangeMeasurement_ind

This indication informs the MAC Client that a range measurement request was received from a remote device, or that the range measurement request performed with a remote device has now completed. The MAC generates this indication when it receives a range measurement command frame from a remote device with Range Type set to Range Measurement Request, or once it has sent a range measurement report to a remote device. The indication therefore advises the client that a range measurement operation has started or stopped.

Message Parameters

Name	Type	Description
SrcEUI	tEUI48	Identifies the EUI of the remote device from which the range measurement request was received.
u8RMN	u8	Number of range measurements requested to be performed.
boStatus	Bool	The status of the measurement transaction. If the range measurement has started: TRUE If the range measurement has completed: FALSE

3.6.3.2.38 Measurement Results Set

The Measurement Results Set is an array of octets. The format of this array is specified in Figure 66 of reference [4], a copy of which is shown in Figure 14.

octets: 1	1	1	4	4	...	4	4
Measurement Count	Range Supported	PHYClockAccuracy	R1C ₁	T2C ₁	...	R1C _N	T2C _N

Figure 14: Measurement Results Set [4]

The Measurement Count specifies the number of measurement results in Measurement Result Set.

The Range Supported field is a bitmap which indicates range measurement support and range measurement precision. Bits are set to ONE to indicate support or to ZERO to indicate the feature is not supported. The value supported should be 0x43 as shown in Table 11.

7	6	5	4	3	2	1	0
Reserved	32-bit counter supported	24-bit counter supported	Sample Precision 4224MHz	Sample Precision 1056MHz	Sample Precision 1056MHz	Sample Precision 528MHz	Range Measurement Supported
0	1	0	0	0	0	1	1

Table 11: Range Supported Field

The PHY Clock Accuracy field indicates the accuracy of the PHY clock in units of ppm. The PHY clock accuracy is specified to be at worst 20 ppm.

The measurement results are provided in pairs of 32-bit Rx and Tx measurements such that;

R1C_n is the range measurement at the reception timer

T2C_n is the range measurement at the transmission timer

etc. etc.

3.6.3.3 MAC Datatypes

This section defines datatypes which are specific to the MAC. These types rely on the Global types and on the TMKii types, which are listed in the following sections.

Datatype	Description
MAC_IDIEType_e	This enumeration defines the literals used in the Identification IE.
MAC_MIBType_e	This type contains an enumeration to allow the MAC_MIB union to be correctly interpreted. It contains a combination of LMAC and UMAC identifiers.
MAC_MIB_pu	This handle provides a reference to a MIB attribute as used in the MLME MIB Get / Set service primitives.
MAC_MIB_u	<p>The LMAC MIB stores the configuration information for the HMAC and the PHY. The LMAC provides services to update and maintain the MIB. Upon HMAC/PHY reset the LMAC uses data in the MIB to initialize them. See [7], [5], [1] for details of the MIB.</p> <p>Some of the read only PHY registers are not yet added:</p> <p>PHY_CRD</p> <p>PHY_PMMODE</p> <p>Other PHY Static Data registers.</p> <p>Requests for setting these will result in an error in the _cfm.</p>
MlmeResultCode_e	This enumeration is provided in MLME confirm messages and are used to indicates the result of a corresponding request.
ScanState_e	This enumeration is provided to set the scan state in the MLME-SCAN primitive.

3.6.3.4 Global Datatypes

The Global datatypes apply to the UWB stack as a whole. This section lists the subset of the Global Data types which are used within the MAC service primitives. Note that these types rely on the TMKii types, which are listed in the following section.

Message Parameter Type	Description
BeaconType_e	The type of beacon reported, indicating whether a received beacon was classified as a neighbour or an alien, or if the beacon was transmitted by the device.
ID128_s	A structure containing 128 bits that can be used in various contexts.
ID24_s	A structure containing 24 bits that can be used in various contexts.
Nonce_s	Structure with 13 octets , used for Nonce in crypto functions.
Reservationbitmactable_s	Structure to define reservation bit map.
tEUI48	A structure containing 6 octets (48 bits). It is intended to hold an Extended Unique Identifier (EUI).
u128	A structure containing 128 bits that can be used in various contexts.

3.6.3.5 TMKii Datatypes

This section gives base datatypes which declared as part of the framework supplied withTMKii.

Message Parameter Type	Description
bool	Boolean flag; False = 0, True = 1
i16	Signed short integer
i32	Signed long integer
i8	Signed char (byte)
u16	Unsigned short integer
u32	Unsigned long integer
u64	unsigned 64-bit value

u8

unsigned char (byte)

3.6.4 Application Interface Description

The HDR ‘Open Platform’ will support an Ethernet based host interface.

The HDR ‘Open Platform’ will support unique EUI and internal IP addresses which will be configured by setting selection switches on the platform, setting the corresponding values as follows:

- 0: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x01 }, {192, 168, 23, 11 }
- 1: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x02 }, {192, 168, 23, 22 }
- 2: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x03 }, {192, 168, 23, 33 }
- 3: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x04 }, {192, 168, 23, 44 }
- 4: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x05 }, {192, 168, 23, 55 }
- 5: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x06 }, {192, 168, 23, 66 }
- 6: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x07 }, {192, 168, 23, 77 }
- 7: { 0x00, 0x11, 0x22, 0x33, 0x44, 0x08 }, {192, 168, 23, 88 }

Following successful configuration of the devices network connection, a simple Windows application will be used to manage initialisation of the platform and connection setup, including starting or joining a beacon group, and negotiation of a reservation using the DRP protocol. Once a stream has been created IP connectivity shall exist between the platforms, providing support for IP based applications such as video streaming using VLC media player, Microsoft Remote Desktop, and file transfer using FTP client/server applications.

Details of platform configuration, the management application and the application setup have been provided in the supporting documentation for deliverable D7.4.2a, the (V)HDR platform (HW/SW) provision to different activity clusters.

4 Alternative HDR-platforms

In parallel with development of the HDR “Open Platform”, commercially available UWB development platforms from external providers have also been considered. Such platforms are required to support the requirements for HDR-communication in line with the objectives of WP7. These requirements can be categorised as follows;

General or high-level requirements:

- Platform (MAC + PHY) based on ECMA 368
- WLP-implementation with reasonable effort, therefore the platform has to support an interface to the IP-layer, from which the application can take advantage
- Provide optional possibility to scale down PHY- and MAC-bandwidth for further applications with reduced need of data rate

Detailed requirements for support of WP8a, see [6]

- Two kinds of RF-interfaces to access medium to increase reliability of transmission; proposal of supporting frequency areas below 10 GHz and around 60 GHz based on WiMedia protocol
- In case of critical application (for example flight control), a TCP/IP-network shouldn't be used
- Reliability and latency achieved by the wireless system should be in regions of existing wired applications
- The application has to be usable worldwide, therefore the regulation masks for aircrafts and radio usage in all countries have to be met
- To allow a large number of used OFDM-bands, a Detect-and-Avoid (DAA) mechanism is preferred
- To cover an aircraft deck of a 70-m-cabin, around 500 wireless nodes are necessary
- Data throughput up to 30 Mbps in down-link and up to 20 Mbps in up-link per terminal is necessary; access points have to deal with the amount of data transfer for multiple terminals, which increase the expected throughput enormously
- Protection against intentional and unintentional jamming or hacking and electro magnetic interference (EMI)
- Localisation of wireless devices; maybe in coexistence with optional LDR-platform
- Time difference for in-system-latency should be less than 1 ms (between different loud speakers) up to 10 ms (indications and illumination); time difference between user input and visible result should be less than 100 ms

Note: The detailed requirements for support of WP8c are still to be defined - TBD3

4.1 Staccato platform

Staccato Communications, a US company which is a member of the WiMedia-Alliance, is a UWB silicon and software provider for PC, PC peripherals, Consumer Electronics and Mobile Devices supporting Wireless USB, High-Speed Bluetooth, and WiMedia IP with complete single-chip, all-CMOS solutions.. Staccato offer a number of silicon solutions based on their Ripcord® product range, of which the single chip solutions SC4501 and SC4502 [7] offer interesting features.

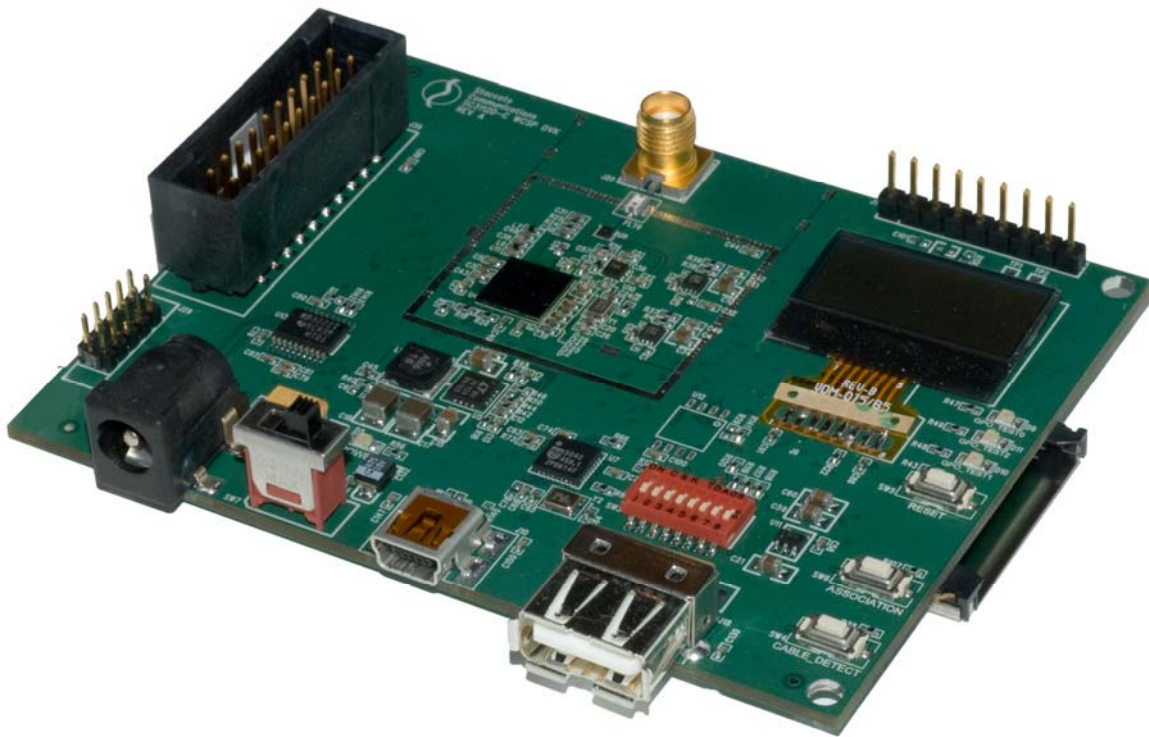


Figure 15 Staccato Communications Ripcord® Development Kit

Currently, discussions and related activities are on going with the hope that this will develop cooperation with Staccato. Unfortunately, this means that the technical details of Staccato's solutions are not presently available to EUWB partners. However, the following points describe those features that are known to be supported.

- Single-chip-solution with WiMedia conform MAC, Baseband and RF-frontend, see Figure 16
- Tiny 5mm x 5mm x 0.55mm package
- WiMedia band groups 1, 3, 6 with DAA-support
- WiMedia TFC 1-10 (TFI and FFI modes)
- All mandatory and optional data rate modes (53.3Mbps up to 480Mbps)
- Flexible MAC-engine operation to support multi-protocol (This feature is a patent-pending-technology designed for gigabit-speed heterogeneous protocol environments.), see Figure 17:

- Wireless USB

- High-speed Bluetooth
- WLP (IP-over-WiMedia)
- Wireless audio/video
- ARM9 processor core
- 128-bit AES hardware engine
- High Performance SDIO 2.0 device interface or USB 2.0 interface available

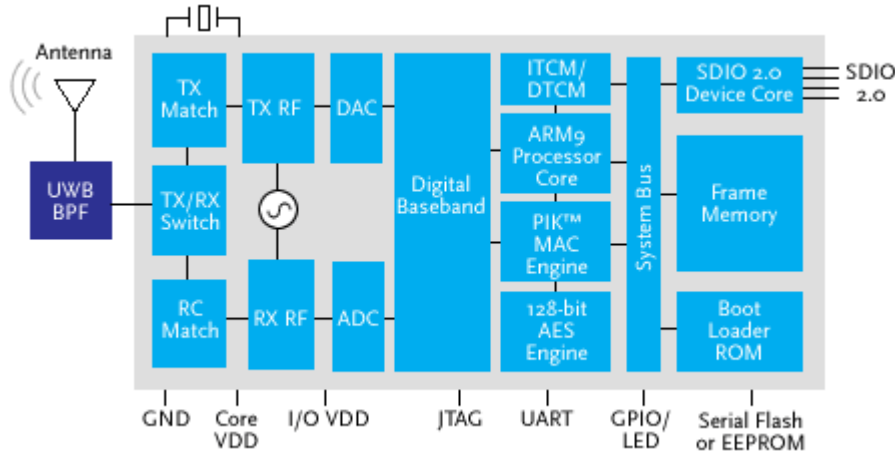


Figure 16: Block diagram of Staccato’s single-chip-solution SC4501 [7]

4.1.1 Development support:

Staccato provides a complete development system, called SC4112D, to give the opportunity to learn about the access and behaviour of the chip SC4501 or SC4502. This toolkit comprises dedicated software libraries and a single-board-computer running embedded Linux to evaluate the MAC- and PHY-characteristics.

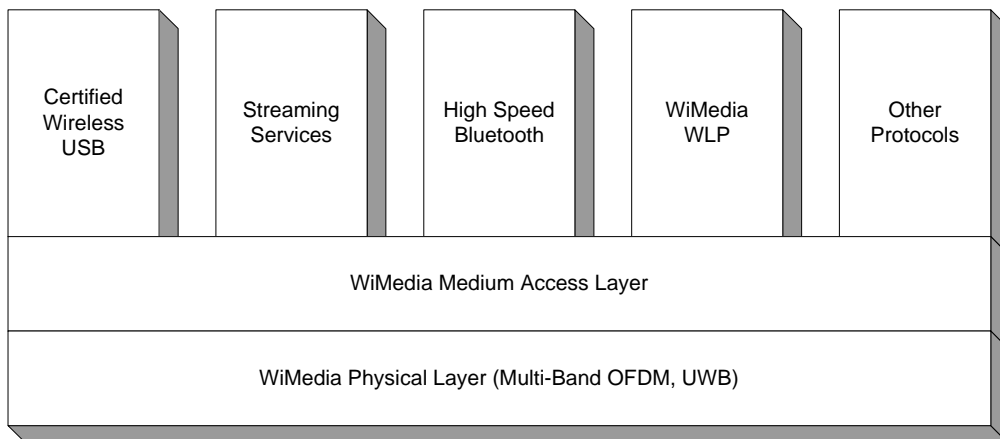


Figure 17: Layer model for Staccatos's multi-protocol-support [7]

4.2 IOGEAR's Wireless USB-solution

IOGEAR, a US company based in Irvine, CA, is a leading connectivity manufacturer that provides complete KVM (Keyboard, Video, Mouse), Connectivity, Networking, Digital Audio/Video, Mobility, and Desktop solutions. IOGEAR provides commercial products based on the WiMedia standard to support Wireless USB connections. Such consumer products are the GUWA100U (Wireless USB host adapter) and the GUWH104 (Wireless USB 4-port-hub), see [8] and Figure 18. It is understood that it is possible to use these devices to establish a wireless linked network, however further investigations are required to ensure that it is possible to support development of a WLP protocol layer using the available device drivers.

The IOGEAR's platform is featured by the following attributes:

- Used frequency band:
 - WiMedia band group 1
 - Frequency range: 3.1 GHz to 4.8 GHz
- Number of channels: 7 (TFI 1-4 and FFI 1-3)
- Supported data rates:
 - 53.3Mbps, 80Mbps, 106.6Mbps, 160Mbps, 200Mbps, 320Mbps, 400Mbps, 480Mbps
- Operating range is up to 5 meters
- USB 2.0 / 1.1 specification interface



Figure 18: IOGEAR's USB host adapter GUWA100U and USB hub GUWH104 [8]

5 Conclusions

This deliverable specifies the design requirements of the HDR “Open Platform” developed in WP7 of the EUWB project. Details of the features supported by the PHY and MAC are provided in sections 3.4 and 3.6 respectively.

The platform provides an integrated solution to include a PHY module, provided by Wisair, and a MAC layer on an FPGA development platform, provided by TES Electronic Solutions. The HDR “Open Platform” will provide a control and data interface via Gb Ethernet to a host PC.

The platform shall support several features which can be used to support development of higher layer protocols and applications. These include:

- An ECMA-368 derived MLME based control interface as specified in section 3.6.3
- An IP based data interface as supported by the host PC driver and application interface described in section 3.6.4

An example application shall be provided to demonstrate these control and data interfaces.

In addition, through the phased development and implementation plan described in section 3.2, the PHY module shall be developed to support operation in band group 4 and localisation and ranging capabilities.

In parallel with specification of the HDR ‘Open Platform’, investigations that have been performed to identify some alternative commercially available development kits and solutions. Section 4 describes two such platforms from Staccato Communications and IOGEAR. It is expected that through support of platforms such as these, EUWB will be able in turn to support a broader range of higher layer protocol and application level developments in supported work-packages.

Currently, there are a small number of specification details which remain outstanding. These have been captured in Appendix I. These details need to be tracked and carefully monitored through development of the HDR “Open Platform” in WP7 in order to clearly communicate the implementation schedule and minimise the impact where these features are required to support developments in other work packages.

Appendix I –Specification Details “TBD” List

Index	ID	Meaning and units	Schedule to resolve	Comments
	TBD 1	Band coverage of next phases implementation [Band group IDs].	Upon selection of PHY Chip for each phase.	Depends on PHY chips proven ability.
	TBD 2	PHY level of support for ranging and localization feature in 3 rd or 4 th implementation stage.	Upon final definition of 3 rd or 4 th implementation stage.	Timing accuracy and access limitations.
	TBD 3	Detect And Avoid implementation	Upon final definition of 3 rd or 4 th implementation stage.	N/A
1.	TBD 4	Based on TBD 2 and 3, still need to explore how DAA and ranging and localization will be supported by the MAC layer, through the PHY provided by Wisair	Upon clarification of timescales and implementation details of TBD 2 and 3 above.	MAC level software requirements for support of ranging should be trivial to implement.
2.	TBD 5	TES are currently evaluating the ML507 board and are still to confirm plans to migrate to this platform for MAC / PHY integration.	Resolved at the end 2008.	Transfer to the ML507 platform is now completed, although there are a small number of technical issues that still need to be resolved.
3.	TBD 6	The definition of detailed requirements for the HDR platform are still to be defined by WP8c. See section 4.	End October 2008	N/A

Table 12 – TBD list

References

- [1] Zeisberg, S., Schreiber, V.: “EUWB - Coexisting Short Range Radio by Advanced Ultra-Wideband RadioTechnology”, ICT Mobile and Wireless Communications Summit, Stockholm, June 2008, accepted for publication
- [2] URL of EUWB consortium: <http://www.euwb.eu>
- [3] MULTIBAND OFDM PHYSICAL LAYER SPECIFICATION, VERSION 1.2 , FEBRUARY 22, 2007
- [4] ECMA-368 Standard 1st Edition, December 2005 “High Rate Ultra Wideband PHY and MAC Standard”
- [5] ECMA-369 Standard 1st Edition, December 2005 “MAC-PHY Interface specification for ECMA-368”
- [6] S. Bovelli, F. Leipold: “EUWB-D8a.1 - Scenario description for public transport applications”
- [7] URL of company Staccato Communications, “www.staccatocommunications.com”, see details for Ripcord2 Single-Chip IC SC4501
- [8] URL of company IOGEAR, “www.iogear.com/product/GUWA100U” for Wireless USB host adapter and “www.iogear.com/product/GUWH104” for Wireless USB 4-port-hub

Acknowledgement

The EUWB consortium would like to acknowledge the support of the European Commission partly funding the EUWB project under Grant Agreement FP7-ICT-215669 [1],[2].