

UNIVERSITÀ DELLA CALABRIA

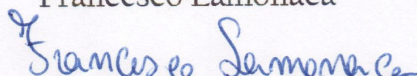
Dottorato di Ricerca in  
Ingegneria dei Sistemi e Informatica

XXII ciclo

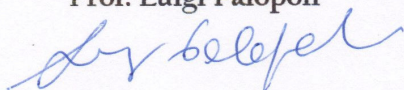
*Tesi di Dottorato*

Delivery the common sense of the time  
to synchronize Measurement Instruments  
co-operating into the DMS

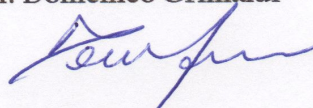
Francesco Lamonaca



Coordinatore  
Prof. Luigi Palopoli



Supervisore  
Prof. Domenico Grimaldi



DIPARTIMENTO DI ELETTRONICA, INFORMATICA E SISTEMISTICA  
Settore Scientifico Disciplinare: ING-INF/07



*To my parents.*



---

## Abstract

*This research is devoted to investigate the coordination in the time domain of the operations executed by the Measurement Instrument (MI) connected to the nodes of the Distributed Measurement System (DMS) by Hardware Interface (HI).*

*On the basis of the synchronization procedures of the node clocks of the DMS presented in the literature, the HIs can work on a synchronized modality. Nevertheless, the hardware and software architecture of the path involved in the communication HI-MI can randomly time delay the commands.*

*Often in the DMS the PC is used as HI. In this case two different aspects are taken into account: (i) the characterization of the hardware connection PC-MI offering the minimum time delay, and (ii) the criteria to modify and to set up the software to reduce the random time delay occurring in the processing steps performed into the PC.*

*The results of the research performed highlights that the main cause of random time delay is the concurrency of the processes running in the PC. In particular, the delay depends on: (i) number of concurrent processes, (ii) their priority, (iii) behavior of the kernel managing the concurrency.*

*In order to overcome the delay caused, the HI based on the Programmable Logic Device (PLD) is taken into account and proposed in the place of the PC. From the analysis of the operations executed on the PLD, operating conditions are shown, making random variation of the synchronization time delay of the MI to the node clock. In order to detect the causes of the random variation, the polling cycle is taken into account and analyzed. From this analysis the model of the uncertainty affecting the synchronization time delay is pointed out in order to evaluate the effects of each cause affecting the trigger check. The result of this evaluation furnishes: (i) information to point out the adequate strategy to reduce the random time variation of the detection of the trigger condition, and (ii) requirements to completely avoid the polling cycle. The experimental tests performed by using implemented embedded HI assess the efficacy of the presented HI to achieve sub-microsecond synchronization accuracy, according to the standard IEEE 1588.*

*In the research the problem of the stand alone MIs is also taken into account. In some cases, indeed, synchronized measurement procedures must be performed in places where the node of the DMS are not reachable, or not convenient to reach, by wired or wireless connections. For this reason it is not possible to use standard synchronization protocol to synchronize the MIs. In these cases the use of the Personal Digital Assistant (PDA) is proposed to physically bring the common sense of time to the stand alone MI. In order to achieve synchronization accuracy in the order of sub-microsecond, it is presented the conjunct use of the PDA and the proper embedded HI. This solution meets the advantages of both the devices. The PDA permits the advantages of the high level programming languages in order to interface the MI and collect the data. The embedded HI guarantees high synchronization accuracy on the basis of the deterministic behavior of the proposed hardware architecture. Experimental tests confirm the suitability of the proposed solution to the requirements of the standard IEEE 1588.*

---

## Sintesi

*L'attività di ricerca è rivolta allo studio del coordinamento nel dominio del tempo delle operazioni eseguite dagli Strumenti di Misura (SM) connessi ai nodi del Sistema Distribuito di Misura (SDM), tramite Interfaccia Hardware (IH).*

*Sulla base delle procedure di sincronizzazione dei nodi del SDM presenti in letteratura, le IH possono operare in modalità sincrona. Tuttavia, le architetture hardware e software del collegamento tra IH e SM possono ritardare i comandi inviati allo SM.*

*Spesso nei SDM il PC è usato quale IH. In questo caso due differenti aspetti sono presi in considerazione: (i) la caratterizzazione della connessione hardware PC-SM che offre il minimo ritardo di comunicazione e (ii) il criterio per modificare e settare il software eseguito dal PC per ridurre l'aleatorietà del ritardo introdotto dall'esecuzione del processo di interfacciamento SM-SDM.*

*I risultati della ricerca hanno evidenziato che la principale causa di aleatorietà del ritardo è la concorrenza tra i processi che condividono le risorse del PC. In particolare, tale ritardo dipende: (i) dal numero dei processi concorrenti, (ii) dalla loro priorità di esecuzione, (iii) dal comportamento del kernel nella gestione della concorrenza. Al fine di ridurre l'influenza di tale causa di ritardo, è stata presa in considerazione l'IH basata su Programmable Logic Device (PLD) e proposta in sostituzione del PC. Dall'analisi delle operazioni eseguite sul PLD, sono state evidenziate condizioni operative che rendono aleatorio il ritardo di sincronizzazione introdotto da questo tipo di interfaccia. Al fine di rilevare tali cause è preso in considerazione ed analizzato il ciclo di polling per la verifica della condizione di trigger, cioè di inizio esecuzione della misura. Sulla base di tale analisi è stato messo a punto il modello matematico dell'incertezza associata al ritardo di sincronizzazione. Cio' ha consentito di valutare l'effetto di ogni causa che rende aleatorio il rilevamento della condizione di trigger. Il risultato di tale analisi fornisce: (i) informazioni per mettere a punto adeguate strategie per ridurre l'aleatorietà del tempo in cui è rilevata la condizione di trigger, e (ii) indicazioni per progettare ed implementare IH dedicata che eviti completamente il ciclo di polling. Test sperimentali*

*dimostrano che l'uso di IH dedicata garantisce livelli di sincronizzazione tra le misure distribuite nell'ordine del sub-microsecondo. Tale risultato soddisfa le specifiche dello standard IEEE 1588.*

*Nella ricerca è stato preso in considerazione anche il caso di SM isolati. Infatti, misure sincronizzate possono essere eseguite in luoghi dove non è presente e/o dove non è conveniente installare una rete di comunicazione wired o wireless. Per tale ragione non è possibile applicare le tecniche di sincronizzazione presenti in letteratura. Per questi casi è proposto l'uso del Personal Digital Assistant (PDA) per trasportare il senso comune di tempo allo SM isolato. Al fine di ottenere livelli di sincronizzazione inferiori del microsecondo, è proposto l'uso congiunto del PDA e IH dedicata. Tale soluzione si avvale dei vantaggi del PDA per quanto riguarda la facilità di realizzazione del software di interfacciamento dello SM tramite l'uso di linguaggi di programmazione di alto livello, e quelli dell' IH dedicata per quanto riguarda l'accuratezza di sincronizzazione. Test sperimentali confermano che la soluzione proposta soddisfa i livelli di accuratezza richiesti dallo standard IEEE 1588.*



---

## Acknowledgments

*I would like to express my gratitude to my tutor, Professor Domenico Grimaldi, for his academic guidance, critiques of my ideas, emotional support, suggestions, and encouragement. My research could not have been finished reasonably without insightful advice from him.*

*Special tanks to my friends En. Domenico Luca Carni and En. Emanuele Garone with whom I spent several time in the laboratory and shared the burden of these years.*

*Thanks to all my friends, they are too many because I can cite all them.*

*Finally, I am deeply grateful to my family for its love, support, and patience during these years. This thesis would not have been possible without its support and confidence. This thesis is dedicated to my parents, Savino and Raffaelina and to my brother Antonio. They always encouraged and believed in me.*



---

# Contents

<b>Abstract</b> .....	VII
<b>Sintesi</b> .....	IX
<b>Acknowledgments</b> .....	XI
<b>Contents</b> .....	XII
<b>List of Figures</b> .....	XV
<b>List of Tables</b> .....	XXII
<b>1 Introduction</b> .....	1
1.1 Motivations and goals .....	1
1.2 Thesis outline .....	5
<b>2 Synchronization Techniques</b> .....	7
2.1 Introduction .....	7
2.2 Techniques to synchronize clocks .....	8
2.3 Synchronization by using GNSS .....	8
2.3.1 GPS .....	9
2.3.2 GLONASS .....	12
2.3.3 GALILEO .....	12
2.3.4 Limits of the synchronization techniques based on GNSS .....	13
2.4 Synchronization based on internet .....	14
2.4.1 NTP .....	14
2.4.2 IEEE 1588 .....	15
2.5 Comparison among synchronization procedures .....	16
2.6 The open questions .....	17
2.7 Conclusions .....	18

<b>3</b>	<b>Delay in the Path PC-MI</b> .....	21
3.1	Introduction .....	21
3.2	Mathematical model of the delay in the path PC-MI .....	22
3.3	Features of the realized test beds .....	24
3.3.1	The reference signal .....	26
3.3.2	Operation performed in the DMS .....	26
3.3.3	Consideration on the trigger event .....	28
3.3.4	Consideration on the OS equipping the client PC .....	29
3.4	Experimental tests to evaluate the connection PC-MI assessing minimum time delay .....	29
3.4.1	First case study: Virtual GPIB and Windows OS .....	30
3.4.2	Second case study: GPIB-USB adapter and Windows OS .....	31
3.4.3	Third case study: GPIB-USB adapter and Linux OS ...	31
3.4.4	Fourth test: External Trigger and Linux OS .....	31
3.4.5	Discussion on the experimental results .....	33
3.5	Experimental tests to evaluate the delay caused by the DSOs .	34
3.6	Delay introduced by PCs .....	36
3.7	Conclusions .....	39
<b>4</b>	<b>Reduction of the Synchronization Time Delay by Linux OS Setting Up</b> .....	41
4.1	Introduction .....	41
4.2	Time delay analysis in the path PC-MI with wireless connection	42
4.3	Setting up of Linux OS .....	43
4.4	Changes in the software driver of the WiFi receiver .....	46
4.5	Performance analysis by Linux OS RT .....	49
4.6	Performance analysis by RTnet .....	51
4.6.1	RTnet .....	51
4.6.2	Lan .....	51
4.6.3	Lan Real Time .....	53
4.6.4	Wireless Lan .....	54
4.6.5	Wireless Lan Real Time .....	55
4.6.6	Discussion on the experimental results .....	57
4.7	Validation of the deterministic OS performances by using different MI .....	58
4.8	Conclusions .....	60
<b>5</b>	<b>Synchronization of MI by Using Industrial Hardware</b> .....	63
5.1	Introduction .....	63
5.2	Interaction between polling cycle and incoming trigger .....	64
5.3	Model of the synchronization time delay .....	65
5.4	Software solutions reducing the execution time of the polling cycle .....	69
5.5	Experimental tests .....	73

5.6	Conclusions .....	75
<b>6</b>	<b>Synchronization of MI by Using Embedded Hardware .....</b>	<b>77</b>
6.1	Introduction .....	77
6.2	Performed operation by proposed HI .....	78
6.3	Synchronization by means of proposed HI .....	80
6.4	Model of random variation of the synchronization time delay by using the proposed HI .....	80
6.5	Actual realization of the proposed HI .....	84
6.6	Experimental tests .....	85
6.7	Conclusions .....	88
<b>7</b>	<b>Deliver the Common Sense of the Time by PDA to Stand Alone MI .....</b>	<b>89</b>
7.1	Introduction .....	89
7.2	First synchronization procedure .....	91
7.3	Outline of the optimized synchronization procedure .....	94
	7.3.1 Synchronization phase .....	94
	7.3.2 Operative phase .....	95
7.4	Working by the optimized synchronization procedure .....	96
	7.4.1 Shoot mode .....	98
7.5	Experimental tests .....	98
	7.5.1 Evaluation of the time stamp accuracy into PDA .....	98
	7.5.2 Accuracy achievable by the two proposed synchronization procedures .....	101
7.6	Conclusions .....	102
<b>8</b>	<b>Synchronizing Stand Alone MI by PDA and Embedded Hardware .....</b>	<b>103</b>
8.1	Introduction .....	103
8.2	Description of the proposed extended HI .....	104
	8.2.1 Theoretical criteria to design ESH .....	104
	8.2.2 Actual realization of the ESH .....	108
8.3	Use of the extended HI .....	109
	8.3.1 First phase: synchronization .....	111
	8.3.2 Second phase: measurement execution .....	111
8.4	Model of the synchronization time delay .....	112
8.5	Experimental tests .....	114
8.6	Conclusions .....	116
	<b>Conclusions and Direction for Future Research .....</b>	<b>117</b>
	<b>References .....</b>	<b>121</b>



---

## List of Figures

2.1	Clock time-keeping ability. . . . .	9
2.2	Block scheme of synchronized DMS. . . . .	17
3.1	Block scheme of the operation executed in the synchronized node of the DMS. . . . .	22
3.2	Block scheme of the Wireless DMS realized in the laboratory to perform the experimental tests. . . . .	25
3.3	Wireless DMS realized in the laboratory to perform the experimental tests. . . . .	25
3.4	Common reference signal sent to the DSO to evaluate the synchronization time delay. . . . .	26
3.5	Block scheme of the operations executed in the DMS. . . . .	27
3.6	Sequence of the operations executed in the DMS. . . . .	28
3.7	Block scheme of the DMS in which the communication between PC-MI is the Virtual-GPIB. . . . .	30
3.8	Block scheme of the DMS in which the communication between PC-MI is the GPIB/USB hardware protocol converter. . . . .	31
3.9	Block scheme of the DMS in which the communication between PC-MI is the GPIB/USB hardware protocol converter and both the slave PCs are equipped by Linux OS. . . . .	32
3.10	Block scheme of the DMS in which the communication between PC-MI is implemented via the External Trigger of the DSO and both the slave PCs are equipped with Linux OS. . . . .	32
3.11	a) Block scheme and b) actual realization of the experimental set up to evaluate the delay caused by the hardware of the two DSOs. . . . .	35
3.12	Trend of the synchronization time delay versus $f_s$ and $f_s/f$ . . . . .	36
3.13	a) Block scheme and b) actual realization of the experimental set up to evaluate the delay between the ramp generated at the pin#2 of the parallel port of each PC. . . . .	37

XVIII List of Figures

3.14	Sequence of the commands planned to evaluate the delay introduced by the PCs.....	38
4.1	Simplified block scheme specifying the delay sources in the communication process with the MI. ....	42
4.2	a) Block scheme and b) actual realization of the experimental set up to evaluate the delay between the ramp generated at the pin#2 of the parallel port of each PC. ....	44
4.3	Sequence of the commands planned to evaluate the delay introduced by the PCs.....	44
4.4	Processing block scheme of the signal received at the WiFi interface. ....	47
4.5	Block scheme of the modified WiFi driver. ....	47
4.6	Experimental set up to evaluate the delay between the ramp generated at the pin#2 of the parallel port of each PC equipped by Linux OS and Linux OS RT. ....	50
4.7	Sequence of the commands planned to evaluate the delay introduced by the operating system Linux OS Knoppix Real Time equipping the two PCs. ....	50
4.8	Experimental set up to evaluate the delay between the two PCs, equipped by Linux OS and the UDP broadcast packet sent by LAN is used as common trigger signal between node. . .	52
4.9	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by LAN and the occupancy of both the CPU nodes is 5%. ....	52
4.10	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by LAN. The occupancy of one CPU is 5% and the other one is 100%. ....	53
4.11	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by RT-LAN and the occupancy of both the CPU nodes is 5%. ....	53
4.12	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by RT-LAN. The occupancy of one CPU is 5% and the other one is 100%. ....	54
4.13	Experimental set up to evaluate the delay between the two PCs, equipped by Linux OS and the UDP broadcast packet sent by WLAN is used as common trigger signal between node.	54



4.14	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by WLAN and the occupancy of both the CPU nodes is 5%. . . . .	55
4.15	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by WLAN. The occupancy of one CPU is 5% and the other one is 100%. . . . .	55
4.16	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by WLAN. The occupancy of one CPU is 5% and the other one is 100%. . . . .	56
4.17	a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by RT-WLAN. The occupancy of one CPU is 5% and the other one is 100%. . . . .	56
4.18	Experimental set up to evaluate the delay between the two PCs, equipped by Linux RT and connected to DSO. . . . .	58
4.19	Experimental set up to evaluate the delay between the two PCs, equipped by Linux RT and connected to AWG. . . . .	60
5.1	a) Polling cycle on PLD#1 and PLD#2, respectively; b) overlap of the previous two polling cycles. . . . .	64
5.2	a) Best case: both the triggers occur after the check; b) worst case: the trigger occurs before the check on PLD#1 and after on PLD#2. . . . .	65
5.3	a) Uncertainty reduction of the incoming trigger can reduce the probability that the worst case, occurs, b) the reduction of the Start Offset can reduce the probability that the worst case, occurs. . . . .	68
5.4	Block scheme of the operations executed by the original control program running into the board Rabbit RCM4400W. . .	69
5.5	Operations performed into the polling cycle. . . . .	70
5.6	Block scheme of the operations executed by the modified control program running into the board Rabbit RCM4400W in order to reduce the time of the polling cycle. . . . .	70
5.7	Operation executed when a) the worst case occurs, b) the best case occurs. . . . .	71

5.8	a) Block scheme of the operations executed by the control program using the modified software driver. b) Block scheme of the operations executed by the control program using the modified software driver in the case the sender signal is used only as wireless Trigger. ....	72
5.9	DMS to evaluate the synchronization time delay introduced by the Rabbit RCM4000W. ....	73
5.10	Sequence of the commands planned to evaluate the delay introduced by the PLDs. ....	73
5.11	a) Trend versus the acquisition time, and b) PDF of the synchronization time delay obtained applying the original program. ....	74
5.12	a) Trend versus the acquisition time, and b) PDF of the synchronization time delay obtained applying the modified program. ....	74
5.13	a) Trend versus the acquisition time, and b) PDF of the synchronization time delay obtained applying the modified software driver. ....	75
6.1	Block scheme of the proposed HI architecture reducing the random time delay variation to check the trigger condition. ....	78
6.2	a) Block scheme of the connection into the Synchronization Phase, and b) block diagram of the performed operation by HI into the Synchronization Phase; c) block scheme of the connection into the Operative Phase for the synchronization of two stand alone MIs, and d) block diagram of the performed operation by HI into the Operative Phase. ....	79
6.3	Delay between the opening of the gate in the counter and the effective start to count. In a) the best case occurs, and the delay is zero. In b) the middle case occurs and the delay is equal to $\frac{1}{f_c} - d_t$ . ....	82
6.4	Block scheme of the HI. ....	84
6.5	Block scheme of the HI and of the Starter Block. ....	84
6.6	Experimental set up to evaluate the delay between the trigger signals generated by two HIs. ....	86
6.7	Snapshot of the DSO to evaluate the delay between the trigger signals generated by the HIs, in the case $f_{c1} = f_{c2}$ . ....	87
6.8	Snapshot of the DSO to evaluate the synchronization time delay between the trigger signals generated by the HIs in the case $f_{c1} \neq f_{c2}$ and a) $n_c = 574 * 10^7$ , b) $n_c = 582 * 10^7$ , c) $n_c = 600 * 10^7$ , d) and $n_c = 607 * 10^7$ . ....	88
7.1	a) PDAs are synchronized to the synchronized node of the DMS, b) the PDAs bring the common sense of time to stand alone MIs. ....	90

7.2	Operations involving the Master PC, the PDA-Slave and the interactions among them, on the basis of the first proposed synchronization procedure. ....	92
7.3	Synchronization and unlock procedures. ....	93
7.4	Block scheme of the interaction among the Master PC and the PDAs in order to perform the synchronization, and to operate in synchronized modality. ....	95
7.5	Synchronization procedure based on look up table and unlock procedures. ....	97
7.6	Interaction among the two PDAs and the Master PC, and measurement of the time delay between the shoot time of the PDAs ....	99
7.7	Planned command sequence to evaluate the delay introduced by the two PDAs. ....	100
7.8	Trend of the mean value and standard deviation of the synchronization time delay between the shoot time of the two PDAs versus the number of cycles (a) without the synchronization, (b) with the first synchronization procedure, and (c) with the optimized synchronization procedure. ....	101
8.1	Timing of the output signal of PDA#1, PDA#2, ESH#1, ESH#2. ....	104
8.2	Timing of the Output signal of PDA#1, PDA#2, ESH#1, ESH#2, HI#1, HI#2. ....	105
8.3	Timing of the output signal of PDA#1, PDA#2, ESH#1, ESH#2, HI#1, HI#2. If the PDAs signals rises one before and the other after the rising signal of the two corresponding ESHs the worst case occurs and the HIs signals delay is equal at most to TS. ....	106
8.4	Block scheme of the proposed ESH architecture. ....	108
8.5	Block scheme of the proposed ESH architecture in the configuration corresponding to a) the First Phase in which the synchronization is performed and b) the second phase in which the HIs are used to interface the MI in order to perform the synchronized measure. ....	109
8.6	Block scheme of the performed operation by the proposed ESH. ....	110
8.7	Delay between the opening of the gate in the counter and the effective start to count. In a) the best case occurs, and the delay is zero. In b) the middle case occurs and the delay is equal to $\frac{1}{f_c} - d_t$ . ....	113
8.8	Experimental set up to evaluate the delay between the trigger signals generated by the two HIs. ....	115

8.9 Trend of the mean value (c) and standard deviation (d) of the synchronization delay between the shoot time of the two PDAs applying the correction factor obtained by the lookup table versus the number of cycles. Trend of the mean value (e) and standard deviation (f) of the synchronization delay between the shoot time of the two PDAs and the proposed EHSs versus the number of cycles. .... 115

---

## List of Tables

2.1	COMPARISON AMONG SYNCHRONIZATION PROCEDURES. ....	16
3.1	EXPERIMENTAL RESULTS OBTAINED BY CHANGING OS AND CONNECTION PROTOCOL. ....	33
4.1	MEAN VALUE ( $\mu$ ) AND STANDARD DEVIATION ( $\sigma$ ) OF THE TIME DELAY IN THE TEST OF FIG.4.2 BY DIFFERENT SETTING OF THE OPERATION SYSTEM. ....	45
4.2	EXPERIMENTAL RESULTS BY MODIFYING THE WiFi-USB DRIVER. ....	48
4.3	MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY DIFFERENT LINUX OS AND CPU OCCUPANCY. ....	49
4.4	MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY DIFFERENT LINUX OS, CPU OCCUPANCY AND COMMUNICATION STANDARD. ....	57
4.5	MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY CONSIDERING TDS220 DSO. ....	59
4.6	MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY CONSIDERING TDS7000 DSO. ....	59
4.7	MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY CONSIDERING AWG INSTRUMENTS. ....	60
5.1	EVALUATION OF THE SYNCHRONIZATION TIME DELAY BY USING THE ORIGINAL CONTROL PROGRAM OF THE BOARD RABBIT RCM4400W AND THE MODIFIED ONE. ....	75
6.1	LIST OF THE COMPONENTS USED TO BUILD THE HI ACCORDING TO THE BLOCK SCHEME OF FIG.6.4 AND FIG.6.5. ....	85



## Introduction

### 1.1 Motivations and goals

The synchronization of co-operating processes is a new challenging requirement that emerges in the context of industrial automation, monitoring systems, inspection and management of strategic power plants, computer networks, and sensor networks applications.

The practical requirements of the synchronization, for measurement applications, concern with the correlation in the time domain of the measurements given by independent Measurement Instruments (MIs) connected to the nodes of the Distributed Measurement System (DMS) by proper Hardware Interface (HI). As a consequence, the following steps are involved: (i) implementation of protocols that guaranteeing the accurate synchronized operation at the node of the DMS, (ii) detection of the time delay among independent measurements, and (iii) compensation for the time delay.

The synchronization can be achieved by means of one of the following approaches: (i) sharing the common control signal, or (ii) sharing the common sense of the time.

The accuracy of the synchronization obtained with the first approach is strongly affected by the behavior of the communication network. Such inconvenience can be partially overcome by avoiding the packet collisions into the network.

Otherwise the synchronization based on sharing the common sense of time, because it requires that each node is equipped by the own clock, translates the problem into the minimization of the temporal drift between each node clock and one take as reference. In this case the accuracy strongly depends on the adopted clock synchronization policy.

Among the various procedures to share the common sense of time, those based on the Global Navigation Satellite Systems (GNSS) enable one to achieve, in established conditions, synchronization accuracy in the order of nanoseconds. The accuracy would satisfy the requirements of the measurement applications. In practice, due to the weakness and unreliability of the

satellite signal, relevant accuracy degradations may arise in several practical measurement contexts.

Moreover, the technical requirement to guarantee the free line-of-sight between the receiving antennas and the satellites makes the GNSS-based synchronization methods not suitable for indoor measurement applications. In many applicable scenarios these solutions are not employable or convenient, and synchronization procedures based on communication networks are used.

The Network Time Protocol (NTP), now established as Internet standard protocol, is used to organize and maintain the clock synchronization of the nodes in respect to the time reference. The fundamental advantage of the NTP is the fact that it includes the procedure to compensate the effects of statistical delay variations encountered in wide area networks. The NTP is designed for a large scale distribution network and can be advantageously used for the synchronization with millisecond accuracy. Often this accuracy is not adequate for some measurement applications.

In order to address the needs of the DMS, the standard IEEE 1588 introduce the Precise Time Protocol (PTP), specifically designed for the clock synchronization with microsecond or sub-microsecond accuracy in local networks.

It is worth remarking that this synchronization strategy does not operate on the MI but only on the clock of the HI. As a consequence, this synchronization strategy is not able to guarantee the measure synchronization by itself. In fact, even if the HIs at each node work in synchronized modality, the hardware and the software architecture involved in the communication with the MI can randomly time delay the commands.

The critical task remains to constrain the MI to operate on deterministic synchronized modality with respect to the clock of the HI.

The aim of this research is the characterization and the reduction of the random time delay between the trigger event instant, i.e. the time instant in which the MI should perform the measure and the effective start of the measurement operations by the MI.

Obviously, such analysis strongly depends on the nature of the HI and of the link between the HI and the MI.

In the DMSs many typologies of HIs may be employed, ranging from general purpose PCs to specialized embedded hardware, each one of them characterized by different features concerning with the causes introducing the random time delay. For such a reason suitably strategies for the random time delay reduction must be studied case-by-case accordingly with the adopted HI.

In particular, the following HIs are taken into account:

- General Purpose PC;
- Industrial Hardware;
- Embedded Hardware.



Each one of these HI permits implementation of a different accuracy level of synchronization time delay.

The PC is the typical interface to the DMS node. The main advantages of PC, used as HI, rely in its ability: (i) to use high level programming to drive the MI, (ii) to store high amount of data (hundreds gigabytes of data), (iii) to process the measurement data by means of proper processing capacity or by sharing the computational burdens among PCs belonging to a computer grid.

Fundamental disadvantage introduced by the PC is the time delay of the commands characterized by Gaussian distribution with high value of the standard deviation. The main reason of this behavior is that the procedure driving the MI is executed in concurrency with other processes. These last can be classified as (i) necessary to the PC functionality, e.g. the kernel; (ii) not strictly necessary to the PC functionality, e.g. the graphical user interface; and (iii) unwanted process, for example viruses, spywares, or malwares.

The execution time of each process is established by the kernel that, on the basis of the scheduling policy, satisfies the requests made by the active processes assigning the CPU for an established slot time.

Therefore, the time delay depends on: (i) the number of the concurrent processes, (ii) the scheduling policy, (iii) the kernel behavior.

In order to reduce the random time delay, the proper set-up of the software is the technique that should be examined. This technique consists of (i) the elimination of unnecessary and unwanted processes and (ii) the increase in priority of the ones involved in the drive of the MI (if allowed by the scheduler). Such an approach, even if revealed experimentally to be able to reduce "in practice" the time delays, does not give any theoretical guarantee on the achieved synchronization accuracy.

Another promising technique is based on the idea of translating the interaction between the PC and the MI at lower level than the user space. This technique imposes the low-level re-programming of the drivers of the MI in order to sensibly reduce both the mean and the standard deviation of the time delay avoiding the effects of the concurrence.

Moreover, the implementation of this technique with the real-time kernels can increase the determinism in the time execution of the processes.

Alternatively to the PC, the HI based on industrial hardware, as the Programmable Logic Device (PLD), is characterized by the absence of concurrency between processes, and could permit high level of synchronization accuracy.

Nevertheless, operating conditions of the PLD suggest further reduction of the standard deviation of the synchronization time delay that can occur. In particular, because the polling on condition is chosen as a technique to relieve the trigger event, two different conditions can occur:

- the trigger event occurs before the check operation in all the interfaces;

- the trigger event occurs before the check operation in one device, after in other one;

In the last case the trigger will be detected after the time necessary to execute the polling cycle  $T_P$ . Therefore, by taken into account two PLDs, the synchronization time delay between their operation can vary between zero and  $T_P$ . In order to reduce the synchronization time delay, a promising technique is the reduction of the number of operations executed in the polling cycle. This goal can be achieved by changing the software architecture used by the PLD.

In order to overcome the inconvenience arising from the random time delay of the start time of the MI caused by the polling cycle, a different HI is taken into account to interface the MI to the node of the DMS. The architecture of this HI includes the PLD as wireless interface, and the Embedded Hardware designed and realized to reduce the random variation of the time delay and to avoid the polling cycle to check the trigger condition. In particular, the new HI is able to guarantee that:

- all the computational resources are devoted to detect the trigger event and to drive the MI;
- the trigger condition always occurs before the trigger check;
- the minimum number of operations is required to detect the trigger event and to drive the MI;
- the easy set-up of the parameters concerning the time execution measurement procedure:
  - start time;
  - number of repetitive measures to execute;
  - time period among the measures.

On occasion, practical applications are required to perform synchronized measures among stand alone MIs not connected or not conveniently connectable by wired or wireless link to DMS. In these cases it is not possible to perform the synchronization by using the techniques presented in literature. In order to overcome these limitations it is possible that a Personal Digital Assistant (PDA) can be used to synchronize the operations executed by the stand-alone nodes.

The main idea consists of:

- select one DMS node as time references. The node is connected to the other ones and then synchronized with one of the techniques mentioned above;
- synchronize the PDAs to the reference node;
- use the PDA to bring "physically" the reference time to MI or another DMS inaccessible by the wired or wireless network.

The coarse resolution of the PDA clock is the first problem to be overcome in order to avoid the reduction of the synchronization accuracy.

A valid solution is the use of the system cycle instead of the internal clock to "count" the time. Indeed, presently the PDA is equipped by CPU characterized by working frequency of hundred MHz. Therefore the achievable time resolution is in the order of ten ns.

Such accuracy would satisfy the requirements of most of the measurement applications but, in practice, the CPU working frequency can vary in respect to its nominal value. By definition, the PDA requires different time intervals to count the same number of system cycles.

To overcome this limitation it is proposed to estimate the real working frequency of the CPU and then change the number of system cycles to count in such a way that the same time interval is always employed.

In order to achieve high synchronization accuracy it is proposed that the conjunct use of the PDA and the Embedded Hardware are taken into account. The PDA permits the advantages of the high level programming languages in order to interface the MI and permits collection of the data. The Embedded Hardware permits the synchronization accuracy deriving by the deterministic behavior of the hardware.

Many are the applications requiring synchronized measurement procedures, and many are the operational conditions in which they are performed.

The goal of the research is to furnish theoretical and practical indications to choose a suitable solution in order to satisfy the synchronization accuracy requirements in an effective and efficient way.

## 1.2 Thesis outline

The thesis is organized as follows.

After the first chapter containing the introduction to the research, the second chapter presents a brief overview of the main synchronization techniques. The analysis of their functioning highlights the problem of the degradation of the synchronization between MIs, due to the random time delay introduced by the HI connecting the MI to the node of the DMS.

In the third chapter the experimental characterization of this random time delay is performed by taking into account the PC as HI. The techniques to reduce this random time delay are proposed and experimentally investigated in chapter four.

In the fifth chapter the industrial hardware interface based on PLD is proposed to connect the MI to the node of the DMS. The complete absence of other processes sharing the computational resources, has highlighted other cause that make random the time delay of the software execution: the polling cycle to check the trigger event. Therefore two software procedures are proposed to reduce the standard deviation of the random time delay.

On the basis of the previous considerations, the sixth chapter presents the embedded hardware that avoids or reduces all the causes of delay.

The aim of the seventh chapter is the synchronization of stand-alone MIs, i.e. MIs not reached and not convenient to reach by wired or wireless link. In this case, it is not possible as per the application of the synchronization techniques presented in literature. Therefore, the PDA is used to physically bring the common sense of the time to the stand alone MI.

In order to achieve the same accuracy granted in the case of node of the DMS, in the eighth chapter the conjunct use of the PDA and the Embedded Hardware is proposed.

## Synchronization Techniques for DMS

*Abstract - The brief review of the main synchronization techniques for measurement applications is reported. On the basis of the comparison among them and their practical uses in the Distributed Measurement Instruments (DMS) is highlighted the differences between the synchronization accuracy achieved by the Hardware Interfaces (HIs) connecting the Measurement Instrument (MI) to the node of the DMS, and the ones achievable by the MI. In particular it is highlighted that the HI introduces variable time delay in the communication with the MI.*

### 2.1 Introduction

In the literature the synchronization is a topical question considering that it is an important service in the distributing computer network [80], [11], [84] and in sensor network applications [102], [19], [95], [146]. It operates on the clock of the network nodes.

Today, precise timing is an inherent part of Global Navigation Satellite Systems (GNSS) [61] like the Global Positioning System (GPS) [88], GLONASS and Galileo [143]. These systems provide timing accuracy that, in established conditions, satisfy the requirements of measurement procedures in particular in the industrial contest. However, many are the causes of accuracy degradation that occurs in practical measurement contests.

In the Distributed Measurement System (DMS) the synchronization by network is preferred [28], due to the high stability of the working conditions and the use of preexistent communication structures.

Among the various procedures, the Network Time Protocol (NTP), now established as an internet standard protocol [29], is used to organize and maintain the clock synchronization of the PC to the time computer reference. Fundamental advantage of the NTP for application on DMS is the fact that its protocol includes procedure to compensate for the effects of statistical delay variations encountered in wide area networks and it is suitable for accurate

and high resolution synchronization throughout the Internet. The numerous experimental tests performed show that, as a consequence of the clock drift of the PC, the synchronization interval is an influencing factor of both the accuracy and the stability of the synchronization [30]. Moreover, the NTP is designed for large scale distributed network and can be advantageous used for synchronization with millisecond accuracy.

In order to address the needs of the DMS, in the standard IEEE 1588 [31] a protocol designed for the clock synchronization with microsecond or sub-microsecond accuracy is proposed. In particular, the standard IEEE 1588 provides the protocol Precise Time Protocol (PTP) for synchronization among heterogeneous networked clocks requiring little network bandwidth overhead, processing power, and administrative setup.

Often the Measurement Instrument (MI) is interfaced to the DMS node by the PC and the synchronization procedure operates on the clock, internal or external, of this last. Therefore the PC can work on synchronized modality, but the hardware and software architecture of the path involved in the communication PC-MI can delay the commands.

## 2.2 Techniques to synchronize clocks

Over the years, the timekeeping community has used many different techniques and systems to help them with the task of synchronizing clocks or time transfer. Fig.2.1 shows the performance level of some of these systems. They include the use of:

- terrestrial communications systems, such as television and telephones (MODEMS);
- direct radio broadcasts (WWV [81] and WWVH [82]);
- navigation systems, such as Loran-C and Global Navigation Satellite Systems (GNSS);
- satellite communications systems such as Two-Way Satellite Time Transfer (TWSTT).

As clocks have become more precise and accurate, the timekeeping community has sought more precise and more stable systems to help them with synchronization. For the timekeeping community, GNSS is a significant contributor to solving the traditional problems of timekeeping. It is a reliable source of time and it is a reliable time-transfer system.

## 2.3 Synchronization by using GNSS

The carrier signals broadcasted by Global Navigation Satellite Systems (GNSS) [61] disseminate precise time, time intervals, and frequency on wide

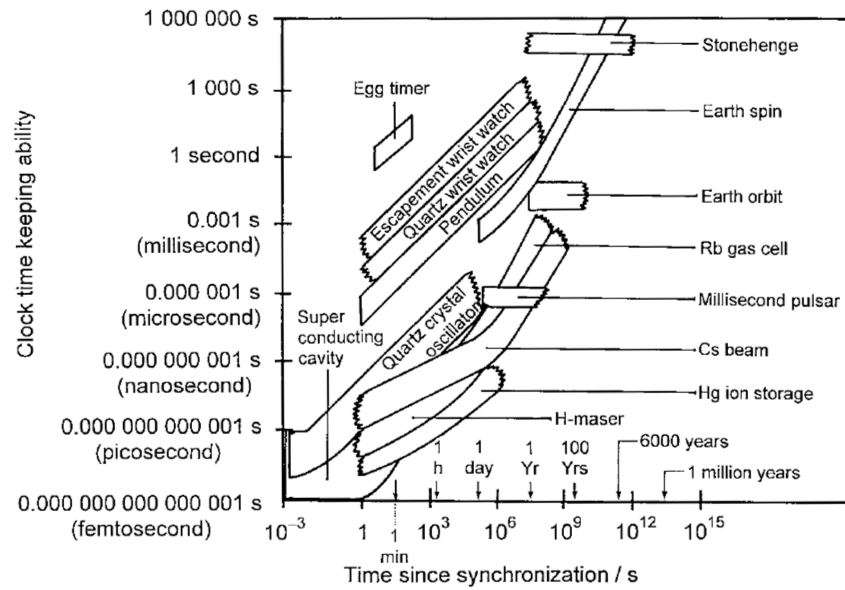


Fig. 2.1: Clock time-keeping ability.

geographic areas. The employment of satellite based timing signals could be particularly suitable since it could make possible the realization of accurate synchronization without requiring the deployment of primary time and time dissemination systems, assuring, at the same time, a set of intrinsic advantages such as wide area coverage, easy access to remote sites and adaptable to changing network patterns. The only user cost is for the receiver equipment, although fees may be levied in future satellite systems.

### 2.3.1 GPS

The Global Positioning System (GPS) [103], [104], [73], [88] is a U.S. space-based global navigation satellite system. It provides continuous reliable positioning, navigation, and timing services to worldwide users [94].

GPS is made up of three parts: between 24 and 32 satellites orbiting the Earth, four control and monitoring stations on Earth, and the GPS receivers owned by users.

GPS satellites broadcast signals from space that are used by GPS receivers to provide three-dimensional location (latitude, longitude, and altitude) plus the time [44].

The satellites are arrayed to provide a minimum worldwide visibility of four satellites at all times. GPS is steered by a ground-based cesium clock ensemble that itself is referenced to Coordinated Universal Time (UTC) [107], [78]. Each

satellite provides a correction to UTC time that the receiver automatically applies to the outputs.

With this continuous adjustment, timing accuracy is limited only by short-term signal reception whose basic accuracy is  $0.2\mu s$ .

This baseline accuracy can be improved by advanced decoding and processing techniques [140].

GPS presently [61] provides two services, one for civilian users referred to as the Standard Positioning Service (SPS) and one available only to authorized users (primarily the U.S. military, and the militaries of U.S. allies) referred to as the Precise Positioning Service (PPS). In 1994, at one time, the accuracy of the SPS was intentionally degraded using a technique referred to as Selective Availability (SA), which was observed to be implemented as a pseudorandom dithering of the satellite clock that could be removed only by PPS receivers with knowledge of the generation algorithm and cryptographic keys [104]. On May 1, 2000, the intentional degradation of SPS performance due to SA was ceased [22]. More recently, in September 2007, the United States announced that the capability to implement SA will be removed from future GPS satellite procurements [105].

Since today, GPS is a navigation system that has proven itself to be a reliable source of positioning for both the military and civilian communities. But it is also an important and valuable utility to the timekeeping community [74].

In the following it is discussed briefly how GPS is used to propagate the common sense of the time.

As part of the navigation solution, the computer contained within the GPS receiver can determine the difference between the clock which is contained within the user receiver and either GPS time or the reference time for GPS, which is UTC (USNO), [44], [107], i.e., UTC as determined at the U.S. Naval Observatory.

The clock contained within a user's GPS receiver is usually a quartz crystal clock. However, in some cases, an external clock such as a rubidium frequency standard or a cesium beam frequency standard can be the local reference for a GPS receiver.

The local receiver can be programmed to display UTC(USNO), as broadcast by GPS, or GPS time since the navigation solution yields the difference between the local receiver clock and GPS time. It should be pointed out that UTC(USNO) is steered to UTC determined by the Bureau International des Poids et Mesures (BIPM) [78]. UTC(USNO) is usually kept to within about 10 ns of UTC.

One advantage the timing community has over the navigation community is the number of GPS satellites that it needs. The navigators need four satellites to determine their position: three to get their position and one to determine the offset of their local clock from GPS time.

Since the timekeepers are in fixed locations and know their position, they only need one GPS satellite to get the offset of their local clock from GPS



time. Consequently, they have modified the algorithms in their timekeeping receivers to take advantage of this fact.

### GPS CV techniques

The GPS Common View(CV) is the techniques with whom GPS Synchronize Clocks over Large Distances.

In the CV technique, two stations simultaneously observe the same GPS satellite. Each of the users at both stations must record the difference between his local clock or local time reference,  $T_a$  and  $T_b$  respectively, and GPS time at the same instant using the same satellite and using a GPS receiver known as a GPS Time-Transfer Unit (TTU). A GPS TTU is a special GPS receiver programmed to compute and display items of interest to the timing community [78].

Therefore, taken into account the two user A and B, it is :

$$User\ A\ observes : A = T_a - GPStime \quad (2.1)$$

$$User\ B\ observes : B = T_b - GPStime \quad (2.2)$$

It is essential that the two users observe the same satellite at the same instant in order to minimize the effects of some errors [6]. By computing the difference between the two sets of numbers:

$$A - B = T_a - GPStime - (T_b - GPStime) = T_a - T_b \quad (2.3)$$

one easily calculates the differences between the two local clocks because the common GPS clock drops out. This is a very simple but powerful process because it is independent of GPS time.

Over the years, significant progress has been made in improving the precision and accuracy of the time distribution and clock synchronization capabilities of GPS. Presently, we are at the 10-25 ns level in one-way time distribution with a coarse acquisition code receiver and at the 2-15 ns level in time synchronization.

A range of values is given for these estimates to indicate that experience has shown that what is achievable is dependent on the user and site specific. These values are representative of the results currently obtained by users in different fields such as telecommunications and metrology.

## Practical application of GPS in DMS

The application of GPS in the DMS was initially strongly limited by the Selected Availability (SA) mechanism [78], removed only in the 2000 [22], and by the high cost of the receivers. Today it is applied to disseminate the sense of the time between DMSs distributed on geographical areas and that can have at least one node equipped by a GPS receiver and that can "see" the sky. A practical application is for example the DMSs applied to monitoring volcanoes, earthquake, changing on the morphology of the territories, [128], etc. In these systems the coordination of the time between the nodes is fundamental to establish, for example, the position of the earthquake epicenter, and the accuracy of the position is depending on the accuracy of the synchronization between the node.

### 2.3.2 GLONASS

The Russian GLObal Navigation Satellite System (GLONASS) was inaugurated in 1982. The full constellation of twenty four satellites plus one spare satellite was completed on 18 January 1996 providing similar capabilities to GPS [77].

Because the GLONASS signal is free of intentional degradation [31] and available world-wide, it offers to the international time metrology community an additional tool for high accuracy time transfer.

However, the GLONASS satellites had a short design life (1 to 3 years [73]) and due to the Russian economical crisis, the constellation has decayed. Fortunately, the GLONASS program has been reinvigorated after that the Russian Federation passed the Decrete Number 587 (August 2001), and in the 2008 consists of only 13 operational satellites [61].

Therefore, because the GLONASS system has not reached its final configuration, today it is used in conjunction with the GPS to improve the timekeeping accuracy [77]. Indeed a single user could see a higher number of satellites, and the modern algorithms implemented in the dual receivers, GPS-GLONASS, improve the accuracy of the position and of the time [18].

### 2.3.3 GALILEO

The European Space Agency (ESA) GALILEO system is the third global satellite time and navigation system to come on line.

It comprises a constellation of 30 satellites divided among three circular orbits at the altitude of 23222 km to cover the Earth's entire surface.

The Galileo system was born as the European answer to GPS and GLONASS, mainly because a return back to the SA techniques damages all the application and research fields strongly depending on the Satellite Time and Frequency Transfer, first of all the European timekeeper community. In [60] it is summarize the Galileo timekeeping infrastructure. The purpose of such infrastructure is to:

- generate a stable time scale (Galileo System Time, GST) to fulfil the needs of the core services (i.e. Open Service, Safety-of-Life Service etc.)
- provide UTC parameters for broadcast
- determine the GPS to Galileo Time Offset (GGTO) in order to facilitate the user's interoperability.

A description of the different time related actors and their corresponding roles within and outside the Galileo core infrastructure was given in [64]. The main elements are the Precise Timing Facility (PTF) [144], [143] being part of the Galileo Ground Mission Segment (GMS), the external Time Service Provider (TSP) [1] and the external GPS system for the provision of GPS to Galileo Time Offset. The core function of the PTF is to generate GST, including a physical realization, GST Master Clock (GST(MC)), for metrological purposes and for providing timing reference signals to all elements within the Galileo system. The main task of the TSP is to provide the GST to UTC mod 1 s steering corrections, and the offset between GST and UTC as required for the provision of the UTC time dissemination service.

Galileo will have an integrity signal to ensure the quality of the signals received and to inform the user immediately of any error. The GALILEO time precision in terms of time errors (95% confidence) for different signals ranges from 0.7-8.1 ns [86].

The higher accuracy granted, the absence of SA mechanism and the automatic detection of satellite errors, make Galileo a new promises for the DMS.

#### 2.3.4 Limits of the synchronization techniques based on GNSS

However many are the causes of not applicability of the synchronization based on satellites for practical DMS applications.

As above mentioned, the hardware receivers need to be reached by the satellite signal, therefore this techniques is not applicable for indoor applications.

Moreover, many are the causes of accuracy degradation in practical contexts. The synchronization based on satellite requires the use of specific hardware to receive and elaborate the signal, and the achievable resolution strongly depends on the quality of this last [18].

Satellite based synchronization systems greatly relies on information transfer over the air interface. This wireless nature of satellite communications links and the weak power levels of GNSS signals make them vulnerable to radio frequency interference. Any electromagnetic radiation source can act as interference source, if it can emit potential radio signals in the satellite signals frequency bands.

The disruption mechanisms that could limit the GNSS performance can be classified as:

- Ionospheric effects: the sunspot activity causes an increase in the solar flux, charged particles and electromagnetic rays emitted from the Sun [101], [65].
- Unintentional Interference: when there will be restricted lines of sight to satellites (i.e. in urban areas, near or under foliage) the synchronization signal quality could deteriorate for short or long term.
- Radio Frequency Interference: It is caused by electronic equipment radiating in the GNSS frequency band (i.e. television/radio broadcast transmitters, mobile phones) [41].
- Intentional Interference: GNSS signals are extremely weak. Therefore they can be deliberately jammed by radio interference. The levels of interference needed to jam a typical consumer GPS receiver are quite low and jamming equipment can be small [41]. Further intentional interferences could be induced by:
  - Spoofing - Counterfeit Signals
  - Meaconing - Delay & Rebroadcast
  - System Damage.

These problems strongly influencing the synchronization accuracy, and therefore can alter the overall measurement processes.

## 2.4 Synchronization based on internet

Synchronizing signals may also be generated from terrestrial systems as far as radio broadcasts [79], [17], [10], microwave, and fiber-optic transmission systems. Nowadays the synchronization over the network is an important research field because it permits (i) to use the persistent and widely diffuse internet network to synchronize computers and embedded systems, and (ii) to overcome the limitation due to the wireless nature of satellite communications links and to the weak power levels of GNSS signals. Therefore, in the following it is reported an overview on the most diffuse synchronization protocol NTP in order to introduce the new standard IEEE 1588 that permits to achieve synchronization accuracy in the order of sub-nanosecond.

### 2.4.1 NTP

The NTP is robust and mature technology for synchronizing a set of network clocks using a set of distributed clients and servers over packet-switched, variable-latency data networks [93]. It is built on the User Datagram Protocol, which provides a connectionless transport mechanism. It is evolved from the Time Protocol and the Internet Control Message Protocol (ICMP) timestamp message and is a suitable replacement for both.

At the top of any NTP hierarchy are one or more reference clocks synchronized to a common time reference and each other using some methods

outside the scope of NTP (i.e. GPS signals, radio signals, or extremely accurate frequency control). Reference clocks are assumed to be accurate. The accuracy of other clocks is judged according to how "close" a clock is to the reference clock, the network latency to the clock, and the claimed accuracy of the clock. It not only corrects the current time, it can keep track of consistent time variations and automatically adjust for time drift on the client. Flexibility of the client/server relationship and security methods allow NTP to work well in many environments and on a wide variety of platforms [29]. The version 4 of the NTP can usually maintain time to within  $10ms$  over the public internet, and can achieve accuracies of  $200\mu s$  or better in local area networks under well established conditions.

#### 2.4.2 IEEE 1588

Standard IEEE 1588 for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems is a protocol [68], often called PTP, used to synchronize real-time clocks in modules of a networked distributed system. It is the evolution of the NTP. Indeed as NTP:

- it is based on the hierarchical organization of the clocks;
- the concept of clock "stratum" is still fundamental to achieve the accuracy of the clock;
- the propagation of the sense of time is implemented by the exchange of time stamped messages in the network.

The protocol was designed specifically to aid in the coordination of activities and the correlation of data in distributed systems typically found in test and measurement, industrial automation, and similar environments. Implementations are easily capable of synchronization to the hundreds of nanosecond range. Sub-nanosecond performance is achievable. Implementations are relatively simple and low cost in both processing and network bandwidth. It is based on two main steps:

- establish the network node clock with the best quality;
- propagate its sense of time to the other clocks of the network.

The first step is fulfilled with the Best Master Clock (BMC) algorithm, the second with the Precision Time Protocol (PTP). The limits of the standard are the hypothesis of:

- the time stamping of the packets is implemented at the egress and ingress of the clock node;
- the symmetry of the communication delay between two nodes.

The problem of the delay fluctuation introduced by the network components, as switch and hub, is overcome by using the Boundary clock, proposed in the version-1 of the standard, and by the transparent clocks proposed in the version-2 presented in March 2008 [69].

This standard best fits the requirements of the DMS because it:

- can be applied in compact systems subnets;
- minimizes the use of the network band;
- minimizes the use of the node resources;
- supports heterogeneous systems of clocks with varying precision, resolution and stability;
- requires low administration;
- needs low cost hardware;
- is applicable to local area networks supporting multicast communications (including but not limited to Ethernet).

Moreover the standard do not give any suggestions on the practical implementation. Therefore in literature hardware, software and hybrid solutions are proposed [72].

## 2.5 Comparison among synchronization procedures

In order to resume the main characteristics of the previous synchronization procedures in Table 2.1 are reported their main characteristics.

**Table 2.1:** COMPARISON AMONG SYNCHRONIZATION PROCEDURES.

	NTP	GNSS	IEEE 1588
Application Area	wide area	wide area	few sub-nets
Communication	internet	satellites	Lan
Accuracy	ms	ns	ns
Administration	configured	n/a	self-organized
Special Hardware	no	receiver	with or without

The synchronization based on satellites permits to achieve synchronization accuracy in the order of ns, cover a wide area but requires the hardware receivers and the accuracy can be corrupted by many different causes due to the intrinsic wireless connection of the communication.

The NTP permits the use of the preexistent internet network but requires human administration and the achieved accuracy is in the order of ms. It depends on the network fluctuation and on the distance of the node clock from the reference clock.

The standard IEEE 1588 matches the advantages of the previous synchronization techniques. It permits to achieve the synchronization in the order of ns, limited to few subnets, as the satellite, and use the preexistent internet network as the NTP.

## 2.6 The open questions

The practical requirements of the synchronization, for measurement applications, concern with the correlation in the time domain of the measurements given by independent Measurement Instruments (MIs) connected to the nodes of the Distributed Measurement System (DMS) by proper Hardware Interface (HI). As a consequence, the following steps are involved: (i) implementation of protocols that guaranteeing the accurate synchronized operation at the node of the DMS, (ii) detection of the time delay among independent measurements, and (iii) compensation for the time delay.

The synchronization can be achieved by means of one of the following approaches: (i) sharing the common control signal, or (ii) sharing the common sense of the time.

Fig.2.2 shows the example of actual realization of the DMS in which the nodes are synchronized by using the standard IEEE 1588.

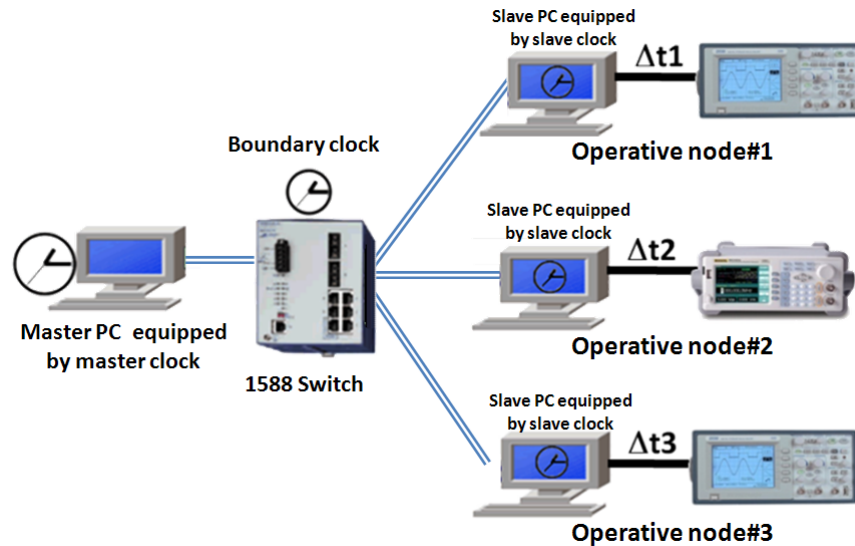


Fig. 2.2: Block scheme of synchronized DMS.

The nodes are equipped by: the Master PC, and three slave PCs. Two slave PCs are equipped by the Digital Storage Oscilloscope (DSO), and one by the Arbitrary Waveform Generator (AWG).

In order to address the needs of the DMS, the standard IEEE 1588 introduce the Precise Time Protocol (PTP), specifically designed for the clock synchronization with microsecond or sub-microsecond accuracy in local networks.

This synchronization strategy, as the other techniques presented in literature, does not operate on the MI but only on the clock of the HI connected to the node of the DMS. In the case of Fig.2.2 the synchronization strategy operates on the clock of the PC connected to each node. As a consequence, this synchronization strategy is not able to guarantee the measure synchronization by itself. In fact, even if the PC at each node work in synchronized modality, the hardware and the software architecture involved in the communication with the MI can randomly time delay the commands. This time is indicated in Fig.2.2 as  $\Delta t_1$ ,  $\Delta t_2$ , and  $\Delta t_3$  respectively in the first, second, and third DMS node.

The critical task remains to constrain the MI to operate on deterministic synchronized modality with respect to the synchronized clock of the PC.

In order to satisfy these requirements the random time delay causes are taken into account and analyzed.

In particular, the path connection between PC-MI involves the hardware connection, the hardware of the MI, and the software procedure of the protocol. Associated to each one, a delay cause can be considered.

Another important delay cause is the software running in the PC devoted to monitoring the event in which the measurement procedure must start.

This event can have different nature according with the particular synchronization process taken into account. In particular, it can be:

- the ring of the synchronized alarm (software or hardware). In this case the synchronization procedure is based on sharing the common sense of the time;
- the changing of voltage level of the common synchronizing electromagnetic signal.

Usually monitoring the trigger event of the polling cycle on the condition is implemented. Therefore a variable time delay is introduced with value distributed between zero and the polling cycle duration.

Moreover, the particular synchronization procedure introduces delay, but in the following it is not taken into account because (i) the generality of the problem taken into account, and (ii) the specialized literature on the synchronization techniques furnish indications on the synchronization accuracy that they achieves.

The goal of this research concerns the characterization and the reduction of the random time delay between the trigger event instant, i.e. the time instant in which the MI should perform the measure, and the effective start of the measurement operations by the MI.

## 2.7 Conclusions

It is presented the scenario of the synchronization procedures that nowadays are interesting the DMS. In particular the ones based on satellite permits



to synchronize in geographical extensions and achieves nanosecond accuracy. The limitations due to the wireless transmission of the signal make it not suitable in many measurement applications. In these cases it is preferred to use synchronization procedures working on the preexistent internet network, as the standard IEEE 1588 that achieves sub-nanosecond accuracy.

The scenario of the synchronization procedures that currently interest the DMS are presented. In particular the scenarios based on satellite permit synchronization in geographical extensions and achieve nanosecond accuracy. The limitations due to the wireless transmission of the signal make it not suitable in many measurement applications. In these cases it is preferred to use synchronization procedures working on the preexistent internet network, as the standard IEEE 1588 that achieves sub-nanosecond accuracy.

All the synchronization procedures work on the clock of the DMS nodes. Nevertheless, in the measurement applications, the synchronization must take into account the time delay of the start of the measurement procedures executed by the MIs in each node. Therefore, the problem of the characterization of the time delay introduced in the path PC-MI is an open question to be investigated.



## Delay in the Path PC-MI

*Abstract - In the previous Chapter it was highlighted that the synchronization techniques operates on the hardware interfaces connecting the Measurement Instruments (MIs) to the Distributed Measurement System (DMS). In the chapter is taken into account that the hardware interface is constituted by the PC. The aim is to characterize the time delay introduced by the path PC-MI and to establish the lower and upper bound.*

### 3.1 Introduction

In the previous Chapter the discussion about the synchronization techniques available in literature and their application in the DMS has highlighted the problem of the time delay introduced by the path PC-MI. This last reduces the accuracy granted by the particular synchronization techniques taken into account to synchronize the DMS nodes.

In order to establish the contribute of the different delay causes, the mathematical model of the delay is taken into account because too many are the causes involved in the behavior of the PC and it is difficult to take into account all them, experimental tests were executed to characterize the time delay and establish the upper and lower bound.

The test beds setting up has features suggested by the mathematical model. In particular the requirements of generality of the results and negligibility of the delay introduced by the chosen synchronization techniques are satisfied.

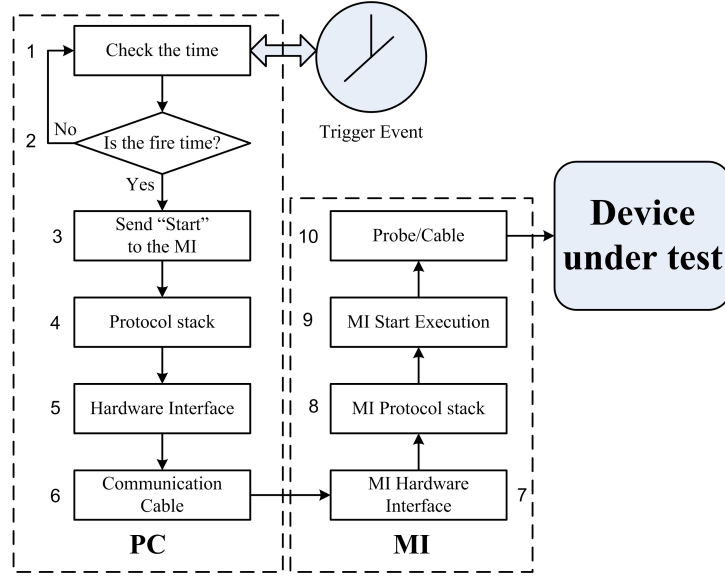
The test beds pointed out permits to establish the influence of:

- the connection PC-MI typically used in the DMS;
- the input section of the MI;
- the Operating System (OS) equipping the PC.

Finally, the results of the experimental tests are discussed in order to furnish indication to reduce the upper bound of the time delay.

### 3.2 Mathematical model of the delay in the path PC-MI

The operations executed by the PC connected to the synchronized node of the DMS and by the MI are shown in the block scheme of Fig.3.1.



**Fig. 3.1:** Block scheme of the operation executed in the synchronized node of the DMS.

The trigger event is the condition that starts the measurement procedure. It can be the fire time of software/hardware alarm, in the case the synchronization between the DMS node is obtained by sharing the common sense of time. Otherwise it can be the incoming of an electromagnetic signal shared between the nodes.

In the follow, in order to highlight the relevance of the problem taken into account, it is supposed that there are no delays between the triggers into the nodes.

Fig.3.1 highlights that some causes of the delay depends by the hardware involved in the path PC-MI. The blocks #5-#10 of Fig.3.1 belongs to this category. The delay introduced by these blocks is characterized by low standard deviation, due to the intrinsic stability of the hardware behavior. Vice versa, the delay introduced by the blocks #1-#4 depends mainly on the behavior of the OS equipping the PC. Indeed the blocks #1-#4 are software executed on general purpose CPU architecture, and the problem of process concurrency must be considered [125]. Therefore, by the block scheme of Fig.3.1, the time

delay  $t_D$  introduced by the connection PC-MI is:

$$t_D = t_{TCT} + t_{EP} + t_{ET} + t_{RD} + t_A + t_P \quad (3.1)$$

where:

- $t_{TCT}$  is the time to check the trigger condition. It depends on the block #1 and #2.
- $t_{EP}$  is the time to execute the procedure to send the Start Acquisition Command (SAC) to the MI. It depends on the block #3.
- $t_{ET}$  is the time to encode and transmit, according to the particular protocol, the SAC to the MI. It depends on the blocks #4, #5 and #6.
- $t_{RD}$  is the time to receive and decode the SAC sent by the PC. It depends on the block #7 and #8.
- $t_A$  is the time to start the execution of the operation on the signal at the MI. It depends on block #9.
- $t_P$  is the time of the propagation of the signal incoming from the observed system in the probe/cable. It depends on block #10.

According to the considerations about the nature of the delay contribute of each block belonging to the path PC-MI, presented in paragraph , it can be assessed that:

- $t_{TCT}$ ,  $t_{EP}$ ,  $t_{ET}$  depend not only on the PC hardware but also on the particular OS that manage it;
- the contribute of  $t_A$  can be considered negligible compared to  $t_{TCT}$ ,  $t_{EP}$ ,  $t_{ET}$  because it depends on the operations executed by the MI;
- $t_P$  can be considered negligible if it is taken into account the length of the most common used probes [66];

Therefore the 3.1 can be rewritten as:

$$t_D = t_{TCT} + t_{EP} + t_{ET} + t_{RD} \quad (3.2)$$

Assuming:

- $f_P$  the Working Frequency (WF) of the PC,
- $f_D$  the WF of the elaboration unit in the MI,
- $n_{TCT}$  the Number of Elementary Operation (NEO) necessary to check the trigger condition,
- $n_{EP}$  the NEO to execute the procedure to send the SAC to the MI,
- $n_{ET}$  the NEO to encode and transmit, according to the particular protocol, the SAC to the MI,
- $n_{RD}$  the NEO to receive and decode the SAC sent by the PC,

the 3.2 can be rewritten as:

$$t_D = n_{TCT}f_p + n_{EP}f_p + n_{ET}f_p + t_{RD}f_{RD} \quad (3.3)$$

In particular,  $n_{RD}$  can be assumed with uncertainty equal to zero if implemented on dedicated hardware. On the contrary  $n_{TCT}$ ,  $n_{EP}$ ,  $n_{ET}$  change their values because the corresponding processes can be preempted on the CPU by the kernel [123]. Therefore, these procedures require more NEOs to be completed: the NEO required by the interesting procedure in the case it runs on dedicated hardware plus ones of the concurrent processes according to the kernel behavior. Moreover the uncertainty associated to the NEO is very difficult to evaluate [85] and mainly depends on [123]:

- behavior of the particular kernel, pre-emptive, not pre-emptive, real time or not;
- number of concurrent processes;
- priority of the concurrent processes;
- particular implementation of the interesting procedure.

For these reasons in order to furnish information about the delay introduced by the Hardware and Software connection PC-MI, experimental tests were performed.

### 3.3 Features of the realized test beds

The DMS pointed out to characterize the synchronization time delay introduced by the path PC-MI satisfies the following requirements:

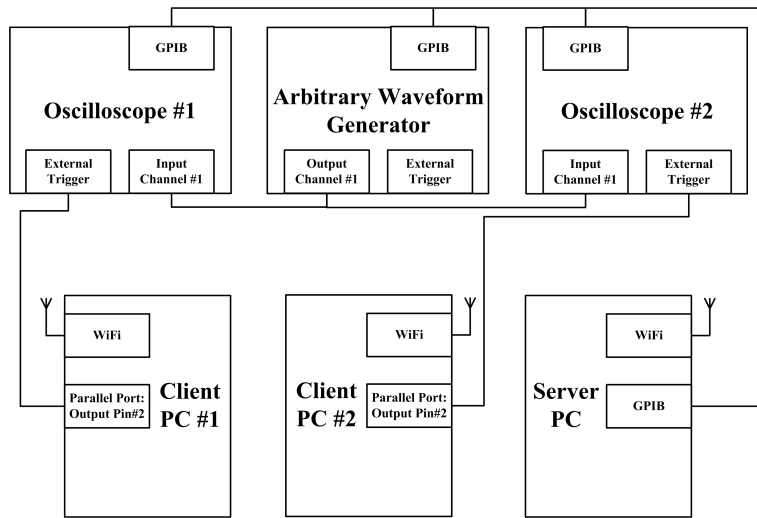
- the DMS can be used in practical cases;
- the chosen synchronization techniques must introduce a delay negligible compared with the one introduced by the connection PC-MI;
- it permits to test different connections PC-MI.
- it permits to test different OSs.

The block scheme of the proposed DMS is shown in Fig.3.2.

The structure of the client node is general and can be split in three parts to analyze the synchronization time delay: (i) the connection PC-MI, (ii) the MI, and (iii) the PC.

In Fig.3.3 is shown the actual realization of the Wireless DMS implemented in the laboratory to perform the experimental tests. There are no difference between the actual realization and the block scheme. Indeed, it is used as MI the Digital Storage Oscilloscope (DSO), Tektronix TDS7404B, that is furnished of PC section. The choice of this particular MI, has permitted to take into account the performance of the Virtual GPIB [127].

As shown in Fig.3.3, the DMS is composed by three nodes.



**Fig. 3.2:** Block scheme of the Wireless DMS realized in the laboratory to perform the experimental tests.



**Fig. 3.3:** Wireless DMS realized in the laboratory to perform the experimental tests.

The first is the master node, equipped by PC that (i) sends the commands to the client nodes, (ii) receives and analyzes the data. The later is the client node, equipped by PC interfaced to the MI.

The choice of the DSO as MI in each slave node is justified because: (i) it furnishes information on the time evolution of the signal, (ii) it can be

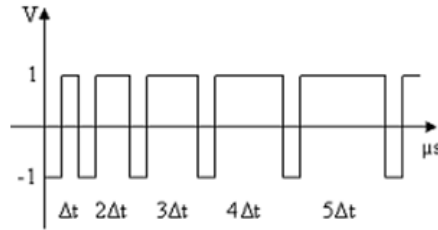
interfaced to the client PC with many different protocols, and (iii) it permits the use of the external trigger to start the acquisition.

This DMS can be used in many practical applications, for example the synchronized monitoring of the signal propagating into a network or into a power network.

In the tests the input signal to the MI is well known and it is used as temporal reference to measure the synchronization time delay.

### 3.3.1 The reference signal

The common reference signal sent to the oscilloscopes is shown in Fig.3.4. The signal is composed by the succession of square waves of different lengths.



**Fig. 3.4:** Common reference signal sent to the DSO to evaluate the synchronization time delay.

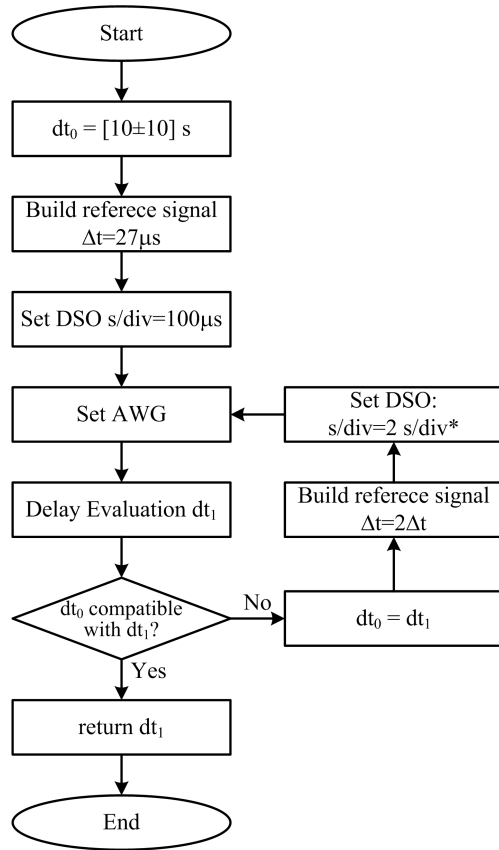
The first wave length is equal to  $\Delta t$ , the  $n$ -th length is equal to  $n\Delta t$ . The amplitude is in the range  $[1V, -1V]$ . The wave form characterized by different length in time of each square wave permits to highlight the time in which each DSO starts the acquisition.

### 3.3.2 Operation performed in the DMS

The software, running in the Master PC, is realized in LabView. The operations executed in the DMS are described in the block diagram shown in Fig.3.5. The procedure pointed out takes into account also the misvaluation due to the periodicity of the common reference signal.

Indeed, also choosing a wide time window on the DSOs by setting the  $s/div$  values, for example equal to  $k$ -time the maximum value of the synchronization time delay observed in preliminary tests, the time delay bigger than the temporal window must taken into account. In this case, the measured time delay is equal to the remainder of the division between the real value of the time delay and the fixed temporal window. It is to highlight that the system and the human operator cannot relive the mistake. Moreover in the DSO a bigger time window reduce the accuracy of the measure.





\*According to the time scale of the DSO

Fig. 3.5: Block scheme of the operations executed in the DMS.

The proposed strategy is to measure the delay with different temporal windows. Therefore the evaluation of the delay is repeated by setting different time scale on the DSOs, and according to these values, it is chosen the proper values of  $\Delta t$  to generate the reference signal.

The delay value is obtained when increasing the temporal window, compatible measures are obtained.

The operations executed to evaluate the synchronization time delay are shown in the time diagram of Fig.3.6. The delay evaluation is performed by executing the correlation algorithm between the signals acquired by the DSOs.

In order to take into account the different working conditions of the PC (automatic activation and suspension of thread, daemon and other processes

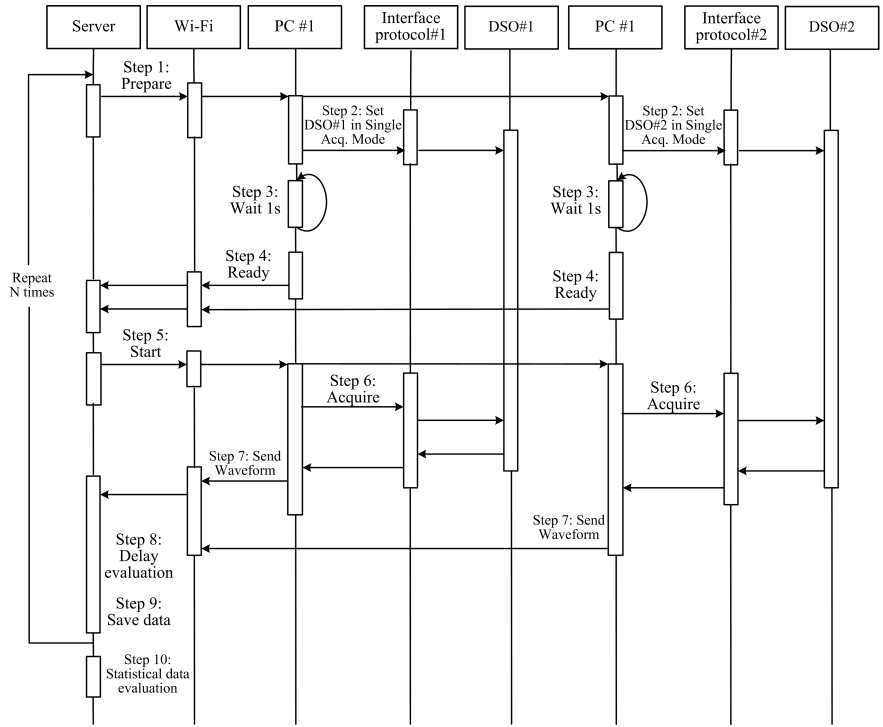


Fig. 3.6: Sequence of the operations executed in the DMS.

in the OS) and to give information about the average value of the delay, each test is executed 4000 times with a periodicity of about 5s.

**3.3.3 Consideration on the trigger event**

The trigger chosen in the DMS is the common electromagnetic signal shared between the client nodes.

In order to reduce the time delay fluctuation in the network components and to make sure that both the slaves receive the command with the same delay, that can be considered negligible with respect to the overall delay taken into account, the following operating conditions are imposed:(i) the transmission of the command from the master to the slaves uses the broadcast ad-hoc connection, and (ii) the communication on the predefined channel is imposed to avoid the time delay that occurs if the slave must search the signal on all the possible channels.

Moreover, the UDP protocol is used in place of the TCP to reduce the delay [67]. The aspects concerning with the packet loss in the UDP is taken into account by introducing intelligent detection and management procedure

on the basis of the experimental architecture. As shown in Fig.3.6, the loose of the UDP packet can be easily detected by the PC server because it does not receive back the answers by the PC clients.

### 3.3.4 Consideration on the OS equipping the client PC

The proposed DMS takes into account also MIs that implement the PC section. Moreover it offers also the possibility to test different OSs.

Indeed, the OS implemented into the MIs is, usually, the Windows OS, and it is strong limitation for the research purposes.

There are many causes that make other OS suitable to interface MI to DMS. Windows is the most diffuse OS and for this reason the biggest part of the viruses are projected to attack it [14], [13]. It is mean that if a virus is in the system it increase the standard deviation of the time execution of the control process because it is an adjunctive concurrent process not controllable by the user. Moreover, the software driver furnished for Windows OS is usually blocked as Windows code. For this reason it is very difficult to improve the commercial code to increase the determinism of the time execution of the control process. Finally, the time resolution of Windows OS is in the order of ms and it does not permit lower synchronization resolution.

## 3.4 Experimental tests to evaluate the connection PC-MI assessing minimum time delay

The MIs and the PC can be equipped by many different communication ports, for examples the GPIB, the serial port, the parallel port, the Lan, the Wireless Lan. Each of them hide the proper communication protocol. The delay introduced by the connection PC-MI is strongly depending on: (i) the complexity of the protocol, and (ii) the number of steps of the protocol implemented via software.

In order to establish the connection that introduce the minimum delay all these types of connection must be suitably tested. Therefore the follow procedure is pointed out:

- build the DMS constituted by two type of nodes, as shown in Fig.3.2:
  - the master node, equipped by PC that sends the commands to the client nodes, receives and analyzes the data.
  - at least two client nodes, equipped by a PC interfaced to the MI;
- the master node sends the common trigger signal to the client nodes. The signal should reach each node at the time  $t$ , characterized by standard deviation that must be negligible with respect to the ones introduced by the client node;
- choose the MI, the HW and SW configuration of the HI for each node;

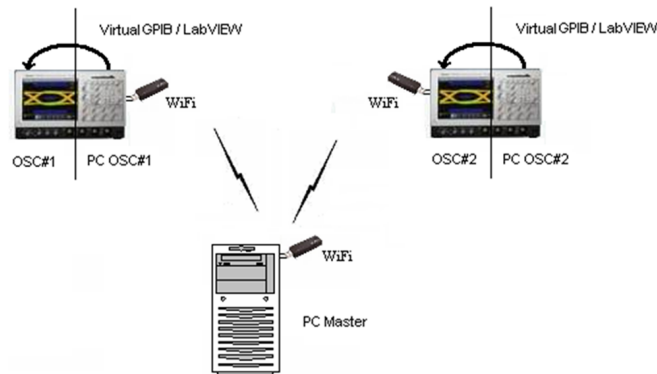
- measure the synchronization time delay between the nodes. The synchronization time delay between the two nodes is the difference between the effective execution time of the command by the MIs. Therefore, it is suggested the use of a common reference signal as temporal references.
- repeat the previous procedure by changing only the connection between PC-MI into the client nodes, in order to test all the connections under examination.

In the follow are shown four representative experimental case studies. They permits to evaluate not only the influence of the particular connection PC-MI, but also the OS running in the client PC.

### 3.4.1 First case study: Virtual GPIB and Windows OS

In the first case study, the DMS realized in the laboratory was composed by: (i) the Master PC PIII 800MHz, with Windows Xp, (ii) the Slave PC#1 realized by the Tektronix TDS7154B equipped with Win2000, (iii) the Slave PC#2 realized by the Tektronix TDS7404B equipped with Win2000, and (iv) wireless communication network realized by IEEE 802.11 (WiFi). Moreover, as shown in Fig.3.3 the Arbitrary Waveform Generator Tektronix AWG420 is connected to the DMS in order to feed both the oscilloscopes with the common test signal.

Moreover, both virtual GPIB and LabView 7.0 are used in the Slave #1 and #2. Fig.3.7 shows the connections realized in the test.

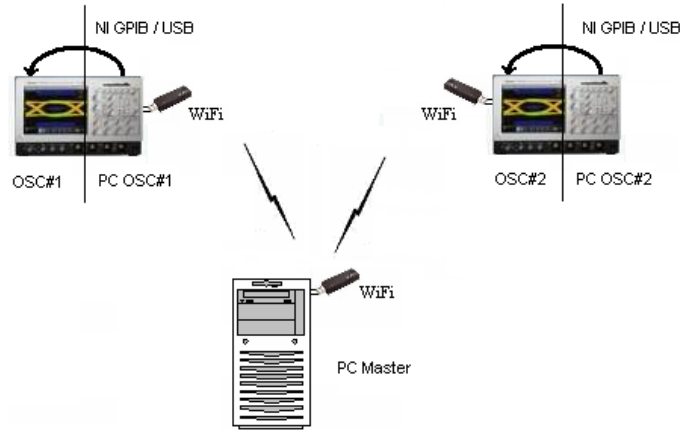


**Fig. 3.7:** Block scheme of the DMS in which the communication between PC-MI is the Virtual-GPIB.

The mean value of the measured synchronization time delay is  $\delta = 0.7384ms$ , the standard deviation is  $\sigma = 0.0127ms$ .

### 3.4.2 Second case study: GPIB-USB adapter and Windows OS

In the second case study, the DMS realized is similar to the previous one, but in the Slave #1 and #2 is used the GPIB/USB hardware protocol converter, instead of the Virtual GPIB. Fig.3.8 shows the connections realized in the test.



**Fig. 3.8:** Block scheme of the DMS in which the communication between PC-MI is the GPIB/USB hardware protocol converter.

The mean value of the measured synchronization time delay is  $\delta = 0.8748ms$ , the standard deviation is  $\sigma = 0.0198ms$ .

### 3.4.3 Third case study: GPIB-USB adapter and Linux OS

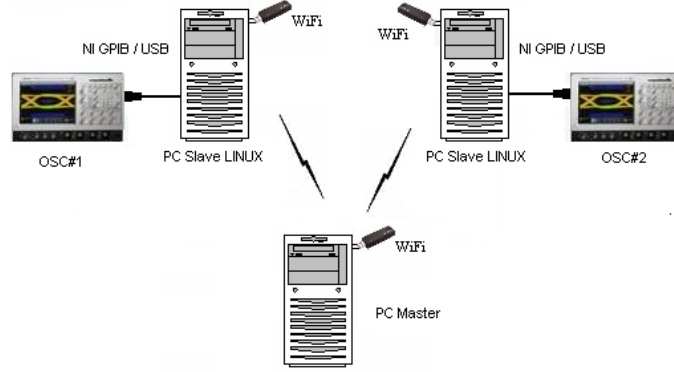
In the third case study, the architecture of the slaves was modified: (i) in the Slave#1 the DSO Tektronix TDS7154B was connected to PC with Linux OS, and (ii) in the Slave#2 the DSO Tektronix TDS7404B was connected to PC with Linux OS. The Linux OS Mandriva 2007 was implemented with the kernel version 2.16.17-5mdv.

Moreover each Slave is connected to the DSO by GPIB/USB hardware protocol converter. Fig.3.9 shows the connections realized in the test.

The mean value of the measured synchronization time delay is  $\delta = 0.4651ms$ , the standard deviation is  $\sigma = 0.0445ms$ .

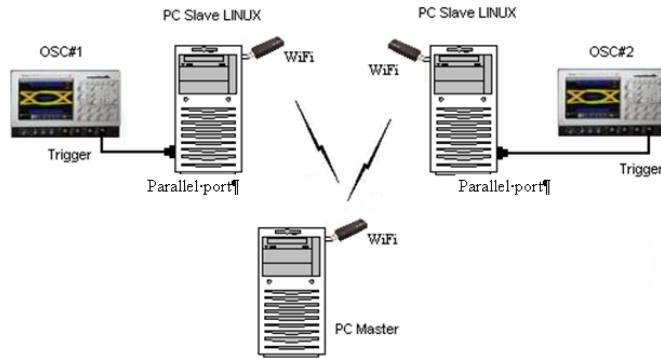
### 3.4.4 Fourth test: External Trigger and Linux OS

In the fourth experimental test, the DMS realized is similar to the ones used in the third test, but the procedure developed in C language controls the



**Fig. 3.9:** Block scheme of the DMS in which the communication between PC-MI is the GPIB/USB hardware protocol converter and both the slave PCs are equipped by Linux OS.

logic state of the pin#2 of the parallel port that is connected to the auxiliary input of the DSO. Fig.3.10 shows the connections realized in the test.



**Fig. 3.10:** Block scheme of the DMS in which the communication between PC-MI is implemented via the External Trigger of the DSO and both the slave PCs are equipped with Linux OS.

The mean value of the synchronization time delay is  $\delta = 0.0523ms$ , the standard deviation is  $\sigma = 0.0058ms$ .

### 3.4.5 Discussion on the experimental results

In order to furnish suggestions about the connection PC-MI assessing lower delay, the obtained experimental results, summarized in Table 3.1, are discussed.

**Table 3.1:** EXPERIMENTAL RESULTS OBTAINED BY CHANGING OS AND CONNECTION PROTOCOL.

		Synchronization Delay (k=1) [ms]
Windows OS	Virtual GPIB	$0.7384 \pm 0.0127$
	GPIB-USB	$0.8748 \pm 0.0198$
Linux OS	GPIB-USB	$0.4651 \pm 0.0445$
	Ext.Trigger	$0.0523 \pm 0.0058$

The experimental results highlight that: comparing the results performed in the first and second experiment, the connection via Virtual GPIB furnishes best results respect to the one GPIB-USB, according to what is assessed by Tektronix [127]. Indeed "virtual GPIB" is a resource which enables to send GPIB-syntax commands via PCI bus from the DSP processor of the digitizer hardware to the system processor of the PC side of the scope [98].

However, the strong limitation of the Virtual GPIB is that it is only available for instruments furnished with PC section.

Comparing the second and the third experiment the standard deviation increases. This is because: (i) it is changed the interface architecture, and (ii) the driver of the GPIB-USB device is changed. Only from few years the NI, producer of the adapter GPIB-USB, is developing product also for Linux OS.

Comparing the third and fourth experiment it is highlighted the drastically improvement of the accuracy, both in mean that in standard deviation.

It is because this solution avoids:

- in the MI the delay due to the execution of the GPIB protocol;
- in the PC the delay due to:
  - the execution time of the driver of the GPIB/USB adapter ;
  - the latency time introduced by the USB port [112].

The executed experimental tests assess that, in order to minimize the time delay due to the connection PC-MI, it is necessary to avoid or minimize the software concurrency. The use of connection that requires complex communication protocol as GPIB, Virtual GPIB increases the influence that the concurrent processes have on the control process [125].

Viceversa the use of the external trigger permits to reduce this influence because avoids communication protocols both in the MI that in the PC. Indeed, in the MI the detection of the trigger event is executed by the acquisition section of the MI, in the PC the number of steps to execute are only 2:

1. check the trigger event;
2. change the logical state of one pin of the parallel port.

On the basis of the proposed solution, based on the use of the external trigger to start the acquisition, the mean value of the achieved synchronization time delay is  $\delta = 0.0523ms$ , the standard deviation is  $\sigma = 0.0058ms$ .

### 3.5 Experimental tests to evaluate the delay caused by the DSOs

The measured time delay takes into account not only the connection PC-MI, but also the delay introduced by the PC and the MI. Therefore, in order to evaluate the delay introduced by the MI, only, the following test is executed.

Fig.3.11a) shows the block scheme, and Fig.3.11b) the actual realization of the connections of the two DSOs, in order to acquire the common signal supplied by the Tektronix AWG420.

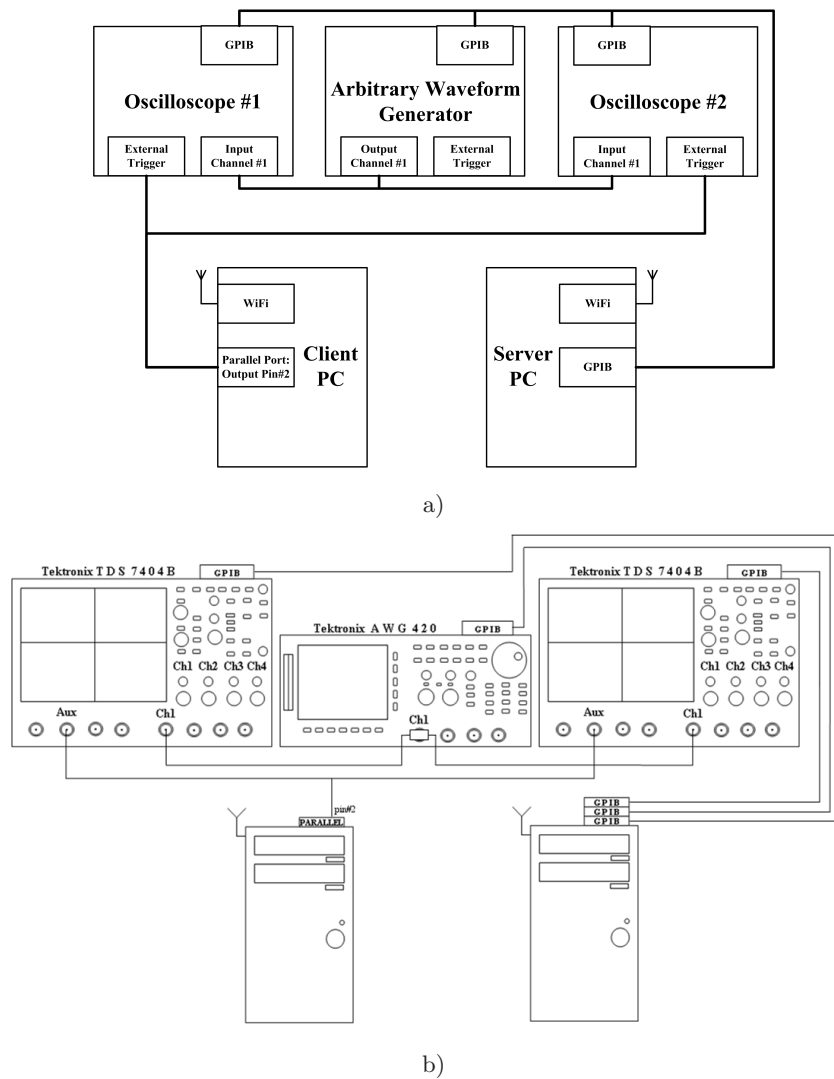
The common trigger at the auxiliary input of each DSO is furnished by the pin#2 of the parallel port of the Slave PC.

The Master PC sends the command to the Slave by WiFi connection and it is connected by GPIB interface bus to the Tektronix AWG420 and the two DSOs. Successively, the signal acquired by each DSO is transferred to the Master by GPIB.

The time delay between these two signals furnishes the evaluation of the delay caused by the hardware of the two DSOs. The procedure based on the numerical algorithm of the cross correlation is used to detect the time delay between the two signals. The cross correlation algorithm processes 4k samples in the case the sampling frequency  $f_s$  is equal to 5GS/s, 10GS/s and 20GS/s, respectively. If the sampling frequency is equal to 0.125GS/s, 0.25GS/s, 0.50GS/s, 1.25GS/s and 2.50GS/s, respectively, the processed samples are 5k.

In order to increase both the sensitivity and the resolution of the cross correlation algorithm, the sinusoidal test signal is employed. In this case the evaluated time delay is introduced by the hardware circuits of the DSOs. Therefore, the use of the sinusoidal signal is acceptable because the expected time delay is low, both in mean value and in standard deviation, respect to the magnitude of the time delay introduced by the software running in the interfaces. The use of different frequencies guarantees against the miscalculation as a consequence of the periodicity of sinusoidal signal.





**Fig. 3.11:** a) Block scheme and b) actual realization of the experimental set up to evaluate the delay caused by the hardware of the two DSOs.

Fig.3.12 shows the trend of the synchronization time delay versus the sampling frequency  $f_s$  and the ratio between the sampling frequency and the frequency  $f$  of the sinusoidal test signal. The maximum delays is lower than 0.2ns.

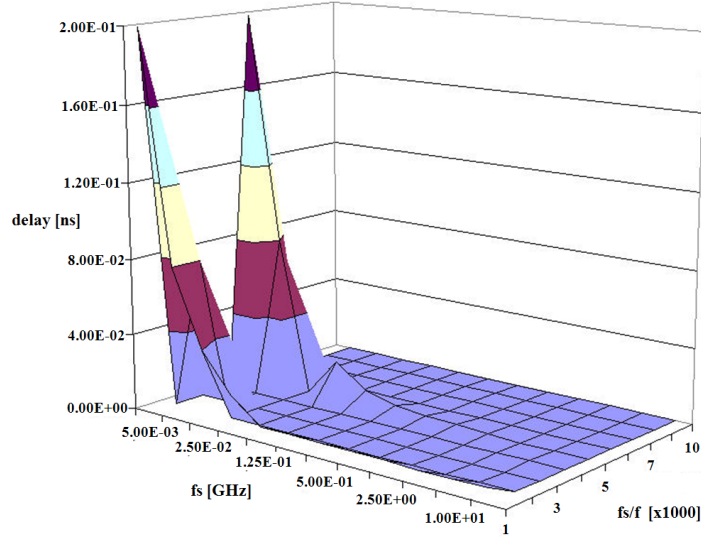


Fig. 3.12: Trend of the synchronization time delay versus  $f_s$  and  $f_s/f$ .

The UDP packet loss is detected by taking into account that (i) the Slave does not send the command to the DSOs, and (ii) the Server receives previous acquired signals.

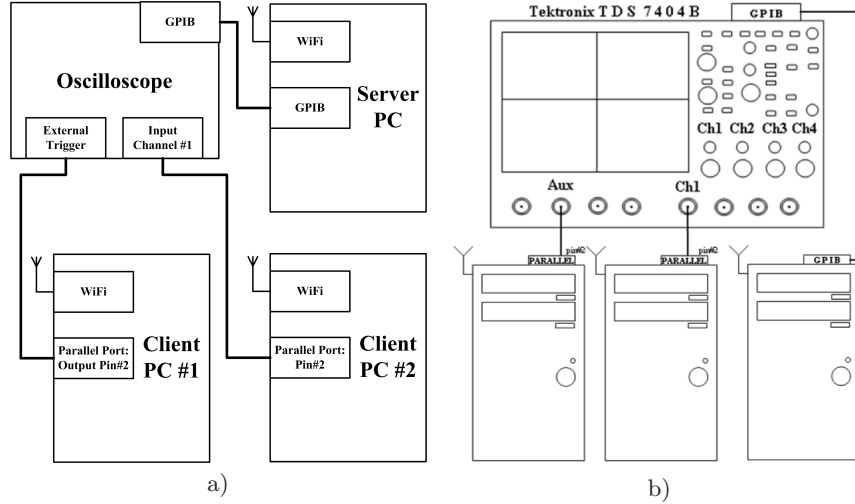
On the basis of the experimental results, the time delay caused by the hardware of the two DSOs must be considered as lower bound.

### 3.6 Delay introduced by PCs

The results obtained in the previous paragraphs suggest that the main influence on the synchronization time delay is the PC. In particular, choosing as connection PC-MI the external trigger of the DSO, the synchronization time delay can be considered depending on the one introduced by the PC.

Indeed, according to the mathematical model presented in the first paragraph, also the term  $t_{RD}$  can be considered negligible because it is depending on the acquisition system of the DSOs, and the term  $t_{ET}$  is strongly reduced because it is the time required to change the logical status of one pin of the parallel port.

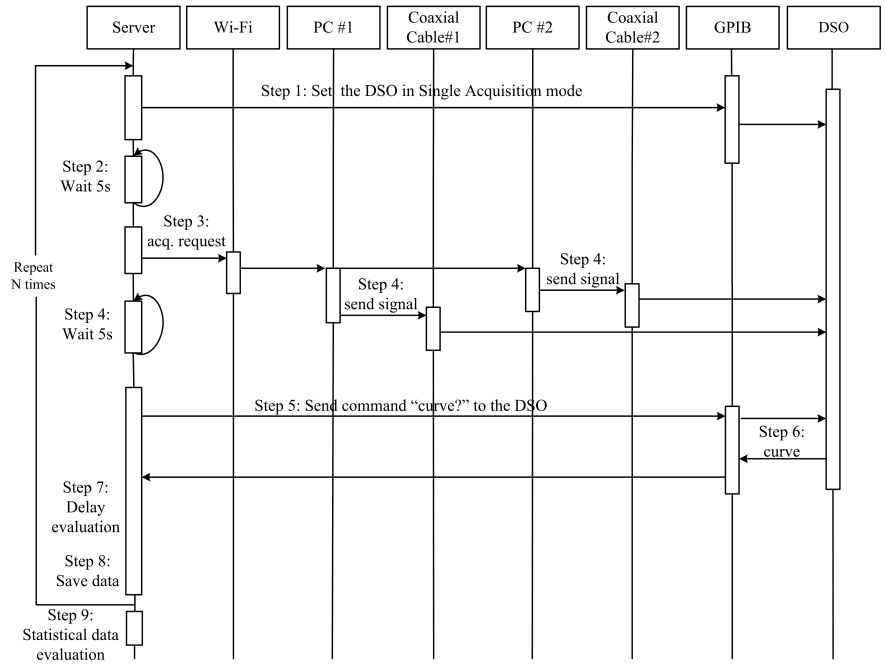
Therefore the test-bed implemented to assess the delay introduced by the PCs can be simplified by using only one DSO instead of two DSO and the arbitrary waveform generator. The block scheme of the new test bed is shown in Fig.3.13a). Its actual realization in Fig.3.13b).



**Fig. 3.13:** a) Block scheme and b) actual realization of the experimental set up to evaluate the delay between the ramp generated at the pin#2 of the parallel port of each PC.

The sequence of the commands and the operations planned to execute the test are shown in Fig.3.14. In particular, the Master PC sends the command to the two Slaves by WiFi connection and it is connected by GPIB interface bus to the DSO. The signal of the Slave#1 is sent to the auxiliary input of the DSO. The signal of the Slave#2 is sent to the channel#1 of the DSO. The signal acquired by the DSO is transferred to the Master by GPIB. The time delay between these two signals furnishes the evaluation of the delay caused by the hardware and the software of the two PCs. The two PCs have similar hardware and OS. The time delay is evaluated by the sample number occurring between the trigger and the cross of the established threshold by the signal acquired. Once set the trigger at 50% of the memory length, it needs to detect the first sample overcoming the threshold, only. In particular, denoted by  $i$  the half of the record length,  $j$  the index corresponding to the sample overcoming the threshold,  $dt$  the time between two samples, the time delay between the trigger and the signal is:

$$\Delta t = |(i - j) * dt| \quad (3.4)$$



**Fig. 3.14:** Sequence of the commands planned to evaluate the delay introduced by the PCs.

Using the previous DMSs the problem of the misevaluation of the time delay due to the periodicity of the common reference signal is solved by repeating the test with different temporal window (see par. 3.3.2). This solution is time consuming and if there are few cases of delay bigger than the temporal window, they are covered by the average operation.

To avoid this misevaluation and to drastically reduce the execution time of the test, the periodic signal, used as common reference signal, is substituted by only one impulse generated to the pin of the parallel port of each PC. The width of the impulse is selectable by the software running in the client PC and it is chosen according to the sampling frequency of the DSO. Indeed if it is lower than the sampling period, it cannot be detected.

The UDP packet loss is detected by taking into account that (i) if only one Slave is involved, it does not send the command to the DSO, and the Server receives the constant signals around the zero value; (ii) if both Slaves are involved the Server receives the previous acquired signal.

The experimental results obtained with the proposed DMS are: the mean value of the synchronization time delay is  $\delta = 0.0458ms$ , the standard deviation of the mean is  $\sigma = 0.0059ms$ .

The result, compatible with the one obtained in the fourth experiment of the paragraph 3.4, confirms the theoretical consideration given in this paragraph and the test bed proposed.

Moreover it is confirmed that, differently from the time delay introduced by the hardware of the two DSOs, the one caused by the PCs must be considered as the upper bound to be reduced.

### 3.7 Conclusions

The aspects concerning with the time synchronization of the co-operating MIs into the DMS are analyzed. In particular, the time delay occurring in the communication between the PC and the DSO to start the acquisition is investigated. The mathematical model of the time delay is pointed out.

It furnish many practical considerations on the causes of the delay and suggests the experimental investigation to evaluate the time delay. Indeed, it is too complicated to study the influence of the PC behavior on the synchronization time delay by using analytical techniques.

The experimental tests have highlighted that:

- the time delay introduced by the hardware component of the path PC-MI is the lower bound of the synchronization time delay and it is in the order of 0.1 ns;
- the time delay introduced by the software executed by hardware general purposes (PC) is the upper bound to be minimized. It is depending on the concurrency of the other processes and on the behavior of the particular kernel;
- the use of connection that requires complex communication protocol as GPIB, Virtual GPIB increases the influence that the concurrent processes have on the control process.

These considerations furnish valid suggestions to further reduce the upper bound of the synchronization time delay. In particular, both the mathematical model and the experimental tests highlighted that to fulfill this goal it is necessary to avoid or minimize the software concurrency.



## Reduction of the Synchronization Time Delay by Linux OS Setting Up

*Abstract - In the previous Chapter it is established that the upper bound of the time delay of the commands occurring in the path PC-MI depends on the OS equipping the PC. The techniques to reduce this delay will be discussed in this Chapter.*

### 4.1 Introduction

Often the PC interfaces the Measurement Instruments (MI) cooperating in the Distributed Measurement System (DMS).

The influence of the Operating System (OS) on the time delay occurring in the path PC-MI was highlighted by the results of the experimental tests discussed in Chapter 3. In particular was shown that the upper bound of the synchronization time delay of the commands occurring in the path PC-MI depends mainly on the execution of the software process to check the trigger and to send the command to the MI. The lower bound depends mainly on the hardware characteristics of the hardware connection between the HI and the MI.

The aim of the Chapter is to reduce the upper bound by analyzing different OSs and their setting up. The proposed solutions concerns with the: (i) setting up of non predictive software Linux OS; (ii) modification of the Linux software driver of the WiFi receiver; (iii) employment of Linux OS RT. These solutions improve the speed and the temporal execution determinism with whom the message received at the interface PC-DMS is processed and transmitted to the MI.

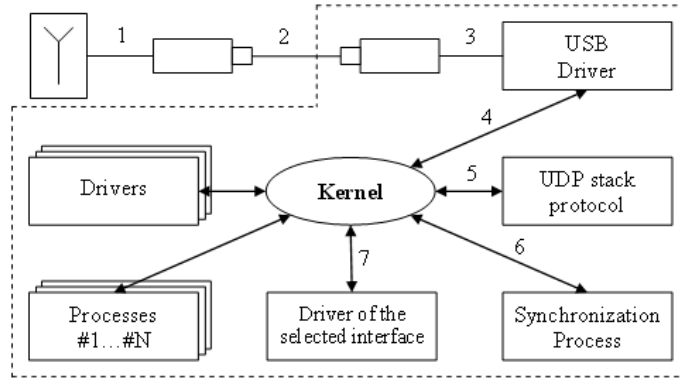
Therefore, the aims is to furnish valid suggestions to achieve the synchronization accuracy required by the measurement procedure with the most efficient solution.

## 4.2 Time delay analysis in the path PC-MI with wireless connection

In order to investigate on the different causes of time delay occurring, the processing steps performed into the PC are taken into account.

In particular, the analysis refers to the Wireless DMS in which the UDP broadcast packet is used as common trigger signal between nodes.

The blocks from 1 to 7, shown in the simplified scheme of the Fig.4.1, are involved in the path inside the PC by the message received at the antenna and switched at the MI [15].



**Fig. 4.1:** Simplified block scheme specifying the delay sources in the communication process with the MI.

The message received at the antenna is hardware decoded and converted in USB compatible signal. The USB software driver decodes the message and transmit it to the UDP stack. Successively, the program implementing the measurement procedure is able to communicate with the MI by the software or hardware interface. The software interface occurs in the modern MI including the PC section in the hardware structure.

In the path inside the PC, two sources of delay can be highlighted. The first one is introduced by the hardware processing of the signal. The second is introduced by the software processing of the message. This last depends on:

- the hardware architecture of the PC,
- the policy of the kernel of the OS,
- both the number and the priority of the processes running at the same time.

In particular, the software runs at the time established by the kernel and another processes can pre-empt the CPU if an interruption occurs. As a consequence, the software processing can be the source of time delay characterized



by high standard deviation, as confirmed by the experimental tests pointed out in the third chapter.

Moreover, the results of the experimental tests highlight the problem to reduce both the mean value and the standard deviation of the synchronization time delay.

The mean value can be reduced by forcing the faster MI to wait the time equal to the estimated mean value. In this case the Linux OS can be advantageously used according to the fact that its time resolution can reach the order of the ns.

The standard deviation can be reduced on the basis of the following strategy: the blocks growing up the standard deviation of the synchronization time delay must have reduced influence in the path inside the PC involved by the message received at the antenna and switched at the MI.

This strategy can be implemented in many different way. Some strategies are easy to implement and does not requires particular experience in the field of the kernel programming and OSs. These solution are suitable in the cases high synchronization accuracy is not required, for example in the monitoring of system with low dynamics.

In the follow different solutions are proposed to reduce the synchronization time delay. They are presented in order of implementation difficulty. For each one the achievable synchronization time accuracy and the critical analysis of the limits and advantages are presented.

### 4.3 Setting up of Linux OS

Linux OS permits to modify the kernel behavior. For example it is possible: (i) kill processes, (ii) change the priority, (iii) remove modules from the kernel.

In this paragraph, the solutions to reduce the synchronization time delay by using only these tools are shown and discussed.

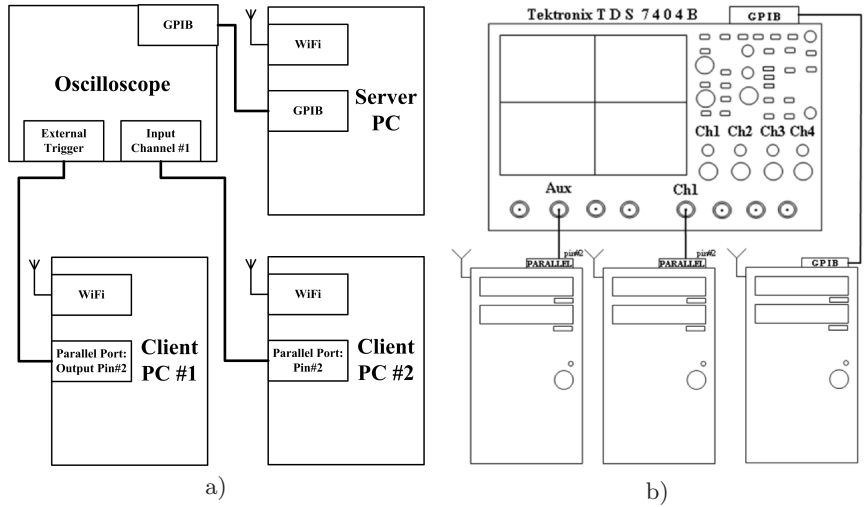
According to the suggestion deriving from the analysis of the mathematical model, discussed in the third chapter, the easy way to validate the proposed strategy is the execution of experimental tests.

Fig.4.2a) shows the block scheme of the DMS used to test the synchronization time delay introduced by the different settings of the Linux OS. Its actual realization is shown in Fig.4.2b).

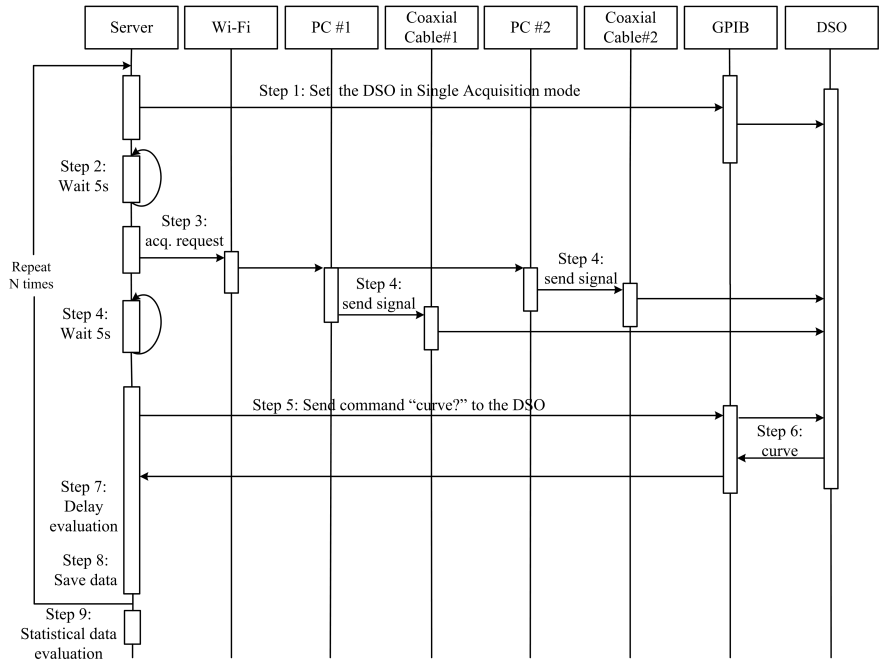
It permits to evaluate the time delay between the signals generated at the pin#2 of the parallel port of each PC.

The sequence of the commands and the operations planned to execute the tests are shown in Fig.4.3.

In particular, the PC Master sends the command to the two Slaves by WiFi connection and it is connected by GPIB interface bus to the DSO. The signal of the Slave#1 is sent to the auxiliary input of the DSO. The signal of the Slave#2 is sent to the channel#1 of the DSO. The signal acquired by the DSO is transferred to the Master by GPIB. The delay between these two



**Fig. 4.2:** a) Block scheme and b) actual realization of the experimental set up to evaluate the delay between the ramp generated at the pin#2 of the parallel port of each PC.



**Fig. 4.3:** Sequence of the commands planned to evaluate the delay introduced by the PCs.

signals furnishes the evaluation of the delay caused by the hardware and the software of the two PCs. The two PCs have similar hardware and OS.

The delay is evaluated by the samples number occurring between the trigger and the cross of the established threshold by the signal acquired. Once set the trigger at 50% of the memory length, it needs to detect the first sample overcoming the threshold, only. In particular, denoted by  $i$  the half of the record length,  $j$  the index corresponding to the sample overcoming the threshold,  $dt$  the time between two samples, the time delay between the trigger and the signal is:

$$\Delta t = |(i - j) * dt| \quad (4.1)$$

This technique avoids the misevaluation of the delay due to the use of the periodical signal as temporal reference among the MIs.

The UDP packet loss is detected by taking into account that (i) if only one Slave is involved, it does not send the command to the DSO, and the Server receives the constant signals around the zero value; (ii) if both Slaves are involved the Server receives previous acquired signal.

Table 4.1 shows the mean value ( $\mu$ ) and the standard deviation ( $\sigma$ ) of the time delay evaluated by referring to the experimental test executed to investigate about the different setting up of the Linux OS.

**Table 4.1:** MEAN VALUE ( $\mu$ ) AND STANDARD DEVIATION ( $\sigma$ ) OF THE TIME DELAY IN THE TEST OF FIG.4.2 BY DIFFERENT SETTING OF THE OPERATION SYSTEM.

Graphic Interface	rmmod-a	Priority	Active Services	$\mu$ [s]	$\sigma$ [s]
y	-	0	1	4.58E-05	5.92E-06
y	-	0	2	4.40E-05	5.43E-06
y	-	0	3	4.09E-05	5.06E-06
-	-	0	1	4.22E-05	5.09E-06
-	-	0	2	4.12E-05	5.05E-06
-	-	0	3	4.21E-05	9.49E-07
-	-	-15	3	4.26E-06	9.55E-08
-	-	-20	3	4.19E-06	9.41E-08
-	y	-20	3	4.19E-06	9.35E-08

The number of time delay measures in each test is equal to 4000 acquired each 10 s. These settings permits to evaluate the mean behavior of the PCs.

Among the different setting up of the OS, interesting experimental results are obtained by:

- reducing the concurrent processes, disabling the unused services and daemons,
- increasing the priority of the synchronization process,
- removing the non active modules from the kernel.

In Table 4.1, the active services are grouped in the three different clusters:

- cluster #1: all the default services and daemons active after the installation of Mandriva 2007,
- cluster #2: services concerning the network interfaces and the kernel utilities,
- cluster #3: added active service to the network interface.

The better result is obtained by:

- disabling the graphical interface,
- keeping the active services of the cluster #3,
- giving the highest priority at the synchronization process,
- removing from the kernel all the inactive modules by means of the command `rmmod -a` [45] and [75].

This solution is the easiest to implement and permit to obtain a synchronization in the order of microsecond.

#### 4.4 Changes in the software driver of the WiFi receiver

Differently from the previous one, that is based on the use of features available in Linux OS, another strategy is based on the conjunction of the synchronization process and interface driver PC-DMS in only one.

This strategy requires changes into the standard software driver and can be valid solution because grown up the predictive proprieties of the process inside the PC occurring in the path input/output. The expected result is that the received command at the WiFi receiver is transmitted directly to the MI.

Into the standard driver, the received signal at the WiFi interface is processed at the hardware level, successively at the kernel level and, finally, at the user level. Fig. 4.4 shows the fundamental hardware and software blocks involved in this process. In particular:

- pAdapter: is the communication buffer between hardware and kernel level. It holds the addresses of the hardware buffer and permits different access methods.
- skbuffer: is the interface between kernel and user level, and, consequently, between the driver and the applications. For each external driver, in the particular case the WiFi driver, the socket buffer is established in order to (i) receive packets once checked in the block protocol check, and (ii) forward the information to application running in the user level.

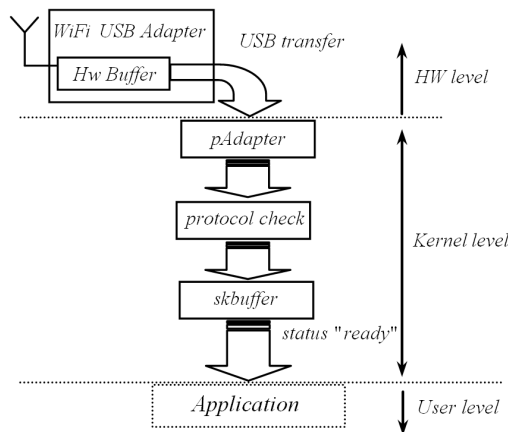


Fig. 4.4: Processing block scheme of the signal received at the WiFi interface.

In order to conjunct the synchronization process and the WiFi interface driver, the command for the MI is plugged in the software driver of the interface. Consequently, the processing software at the kernel level, shown in Fig.4.4, is modified. Fig.4.5 shows the block scheme of the modified WiFi driver in order to permit the introduction of the query in the block protocol check. The scheduler policy of the Linux OS can influence the expected results.

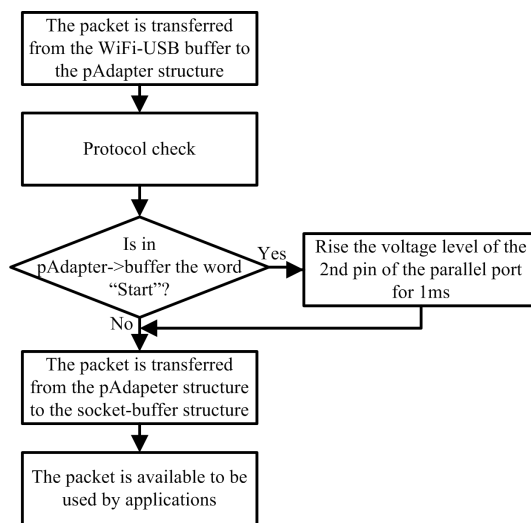


Fig. 4.5: Block scheme of the modified WiFi driver.

Indeed, from the kernel version 2.6 the options pre-empt and not pre-empt are available. The pre-empt option makes interrupt the process running in the case one of the following conditions occur:

- hardware interrupt,
- software interrupt,
- I/O interrupt, requested from the running process,
- time out.

The not pre-empt option operates in opposite manner because avoids to interrupt the running process. Nevertheless, on the basis of the policy of this option, the new process can start only at the end of the previous one.

As a consequence, in all the functioning conditions both the options pre-empt and not pre-empt do not ensure the prompt direct transmission to the MI of the received command at the input WiFi. Therefore, in all the functioning conditions there are delaying causes that can influence the expected results.

Experimental tests were pointed out to investigate about the influence of the options pre-empt and not pre-empt on the performance achievable by using the modified driver to conjunct the synchronization process and the WiFi interface process. The DMS used to test the solution is the one presented in previous paragraph.

Table 4.2 shows the mean value and the standard deviation of the time delay evaluated by referring to different software equipping the PCs. In order to reduce the influence of the scheduler policy, tests are pointed out by increasing the priority of the WiFi USB process.

The experimental results highlight the advantageous use of the modified driver in both the options pre-empt and not pre-empt. Nevertheless, the pre-empt option can be preferred on the basis of the normal working conditions of the PC into the DMS.

**Table 4.2:** EXPERIMENTAL RESULTS BY MODIFYING THE WiFi-USB DRIVER.

Configuration	$\mu [s]$	$\sigma [s]$
PC#1: program; pre-emptive kernel PC#2: program; pre-emptive kernel	4.58E-05	5.92E-06
PC#1: modified driver; pre-emptive kernel PC#2: modified driver; pre-emptive kernel	4.28E-05	3.07E-08
PC#1: modified driver; pre-emptive kernel PC#2: modified driver; pre-emptive kernel Increasing WiFi-USB process priority	4.10E-05	2.89E-08
PC#1: modified driver; NOT pre-emptive kernel PC#2: modified driver; NOT pre-emptive kernel Increasing WiFi-USB process priority	4.16E-05	2.96E-08

## 4.5 Performance analysis by Linux OS RT

Experimental tests were pointed out to investigate about the behavior of the Linux OS Knoppix Real Time, Kernel 2.6 [50].

Fig.4.6 shows the connection among the DSO and the two PCs in order to evaluate the delay between the signal generated at the pin#2 of the parallel port of each PC. The sequence of the commands and operations planned to execute the test are shown in Fig.4.7.

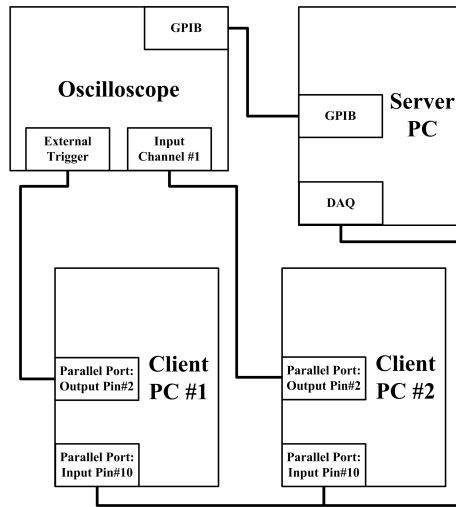
In particular, the PC Master is connected by GPIB interface bus to the DSO. It sends the Start acquisition command to the slaves by changing the logic state of their pin#10 of the parallel port. This change is obtained by using the electromagnetic signal generated by using the DAQ board and shared in parallel modality between the slaves. The signal of the Slave#1 is sent to the auxiliary input of the DSO. The signal of the Slave#2 is sent to the channel#1 of the DSO. The signal acquired by the DSO is transferred to the Master by GPIB. The delay between these two signals represents the delay caused by the software of the two PCs. The two PCs have similar hardware and OS according to the combination given in Table 4.3. In the case both the PCs are equipped by Linux OS RT, the sequence of the planned command previously examined ensures that the running process is able to execute the start acquisition command as soon as possible without influence of concurrent processes. Indeed, the command "ready" alerts the process to control in real time the parallel port. The delay is evaluated in similar manner as in the case of the paragraph 4.4.

From the results of Table 4.3 the following considerations hold:

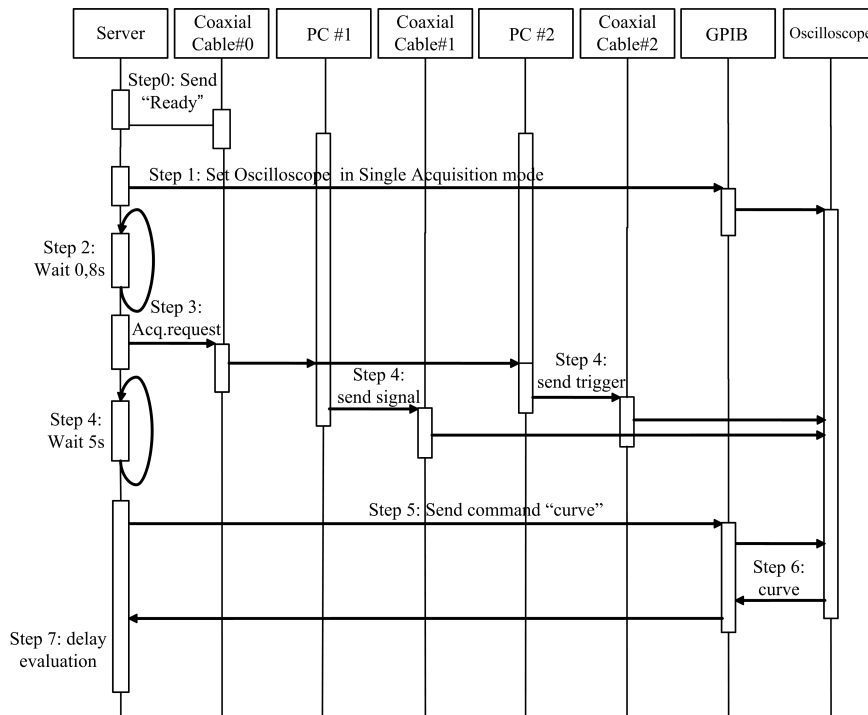
- Linux OS RT operates in independent manner  $\mu$  from the CPU occupancy [50];
- the mean value of the delay and the associated standard deviation are reduced in comparison with the different setting up of Linux OS.

**Table 4.3:** MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY DIFFERENT LINUX OS AND CPU OCCUPANCY.

Operating System	CPU Occupancy		$\mu[s]$	$\sigma[s]$
	PC#1	PC#2		
Linux OS	5%	5%	4.58E-05	5.92E-06
	5%	100%	2.69E-02	1.06E-03
	100%	5%	2.69E-02	1.06E-03
Linux OS RT	5%	5%	4.67E-07	1.03E-08
	5%	100%	4.62E-07	1.02E-08
	100%	5%	4.75E-07	1.04E-08



**Fig. 4.6:** Experimental set up to evaluate the delay between the ramp generated at the pin#2 of the parallel port of each PC equipped by Linux OS and Linux OS RT.



**Fig. 4.7:** Sequence of the commands planned to evaluate the delay introduced by the operating system Linux OS Knoppix Real Time equipping the two PCs.



## 4.6 Performance analysis by RTnet

In the test performed in paragraphs 4.3 and 4.4 the UDP broadcast signal is used as common reference trigger to synchronize the nodes.

In the paragraph 4.5 the common trigger is the electromagnetic signal generated by the master PC by using the Data Acquisition board (DAQ) and sent in parallel modality on the pin#10 of the parallel port of Slave PC.

Thus, it is not possible furnish a comparative evaluation between the obtained results. Indeed the IEEE 802.11 protocol requires a big number of steps to be completed comparing with the changing of the logical state of the pin of the parallel port. The RTAI realtime OS, does not furnish any grants on the time execution of the process if it requires communication on network.

Therefore, in order to give a complete analysis and to assess the advantages of the Real Time OS, new experimental tests are performed. In particular RTnet [116] is used with RTAI to sent the UDP packet in Real Time modality.

### 4.6.1 RTnet

RTnet [116] is Open Source hard real-time network protocol stack for Xenomai [139] and RTAI [115] (real-time Linux extensions). It makes use of standard Ethernet hardware and supports several popular NIC chip sets, including Gigabit Ethernet. Moreover, Ethernet-over-1394 support is available based on the RT-FireWire protocol stack.

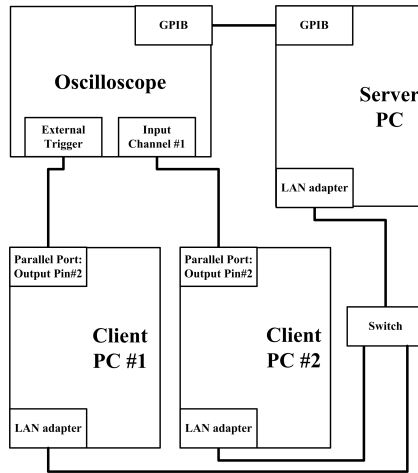
RTnet implements UDP/IP, ICMP and ARP in a deterministic way. It provides a POSIX socket API to real-time user space processes and kernel modules.

To avoid unpredictable collisions and congestions on Ethernet, an additional protocol layer called RTMac controls the media access. A dedicated Ethernet segment is required to guarantee bounded transmission delays, but RTnet also includes a mechanism to tunnel non real-time traffic like TCP/IP over RTMac, thus allowing a "single-cable" solution for connecting control systems.

This proprieties are useful during the execution of the tests because permit on the same test bed the implementation of both the Real Time and standard communication. Therefore, it is possible to compare the experimental results because obtained by using the same hardware architecture changing only the OS and in the network protocol stack.

### 4.6.2 Lan

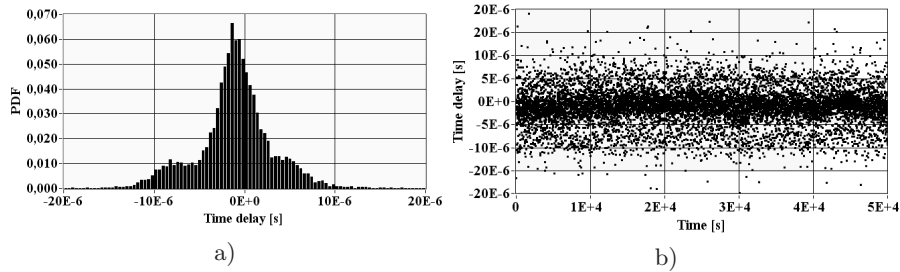
Fig.4.8 shows the block scheme of the DMS used to evaluate the synchronization time delay in the case the synchronization trigger used in the node is the incoming UDP broadcast signal sent by LAN.



**Fig. 4.8:** Experimental set up to evaluate the delay between the two PCs, equipped by Linux OS and the UDP broadcast packet sent by LAN is used as common trigger signal between node.

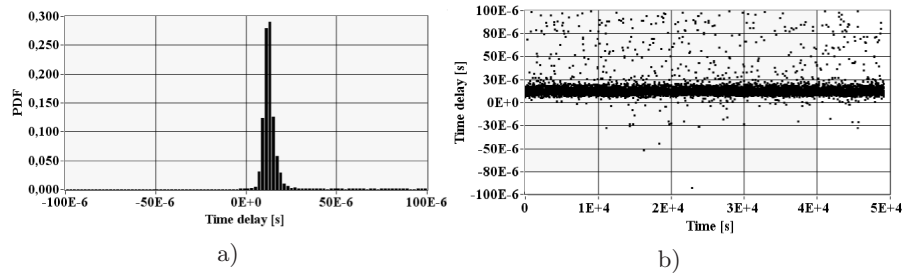
In the DMS the switch is used to reduce the influence of the network device on the evaluated time delay. The operations performed to evaluate the time delay are the same described in the paragraph 4.5.

Similarly to the procedures executed in paragraph 4.5 it is also tested the trend of the synchronization time delay in the case of different CPU occupancy. Fig.4.9a) shows the Probability Density Function (PDF) and Fig.4.9b) the trend versus the time of the Synchronization time delay in the case the CPU works at 5% of occupancy.



**Fig. 4.9:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by LAN and the occupancy of both the CPU nodes is 5%.

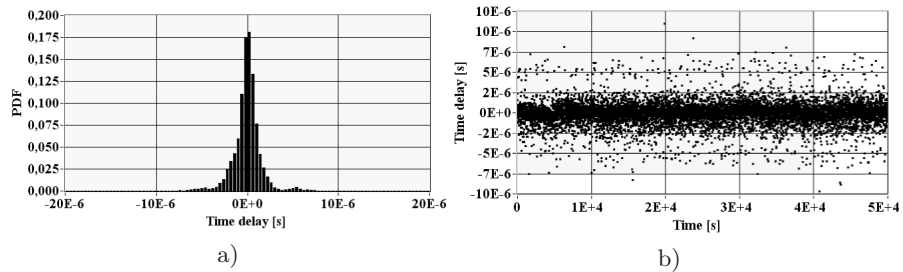
In Fig.4.10a) is shown the Probability Density Function (PDF) and in Fig.4.10b) the trend versus the time of the evaluated synchronization time delay in the case the CPU works one at 5% and the other one at 100% of occupancy.



**Fig. 4.10:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by LAN. The occupancy of one CPU is 5% and the other one is 100%.

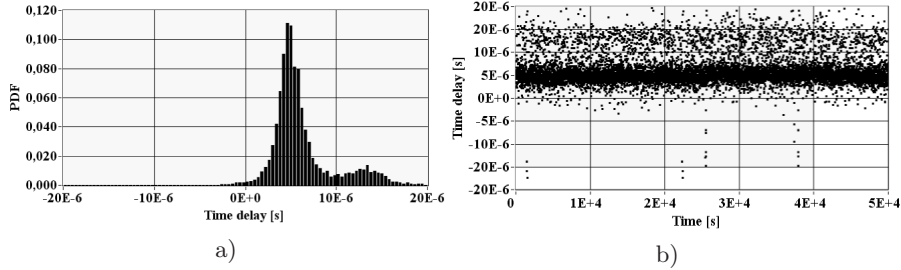
#### 4.6.3 Lan Real Time

In this test, the DMS pointed out is similar to the ones used in paragraph 4.6.2, but in the Slave #1 and #2 is used the Real Time OS and RTnet. Fig.4.11a) shows the PDF and Fig.4.11b) the trend of the synchronization time delay in the case both the CPU work with the same occupancy equal to 5% versus time.



**Fig. 4.11:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by RT-LAN and the occupancy of both the CPU nodes is 5%.

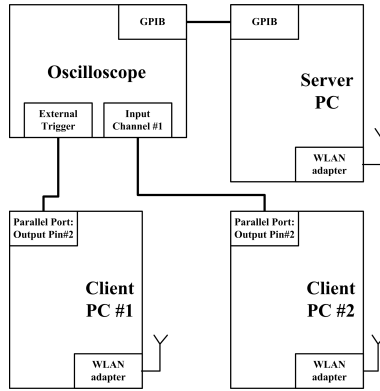
Fig.4.12a) shows the PDF and Fig.4.12b) the trend versus the time of the synchronization time delay in the case one CPU work with occupancy equal to 5%, the other one with occupancy equal to 100%.



**Fig. 4.12:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by RT-LAN. The occupancy of one CPU is 5% and the other one is 100%.

#### 4.6.4 Wireless Lan

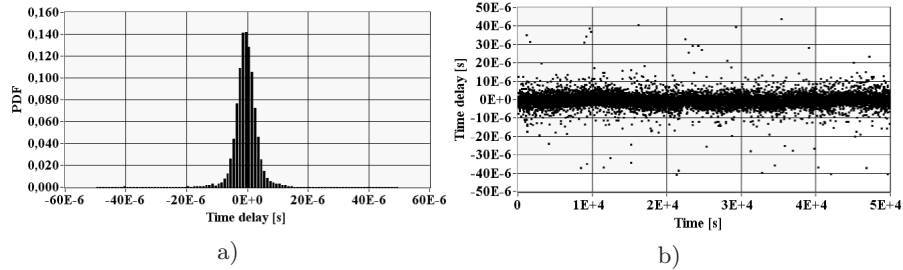
Fig.4.13 shows the block scheme of the DMS used to evaluate the synchronization time delay in the case the synchronization trigger used in the node is the incoming of the UDP broadcast signal sent by WLAN.



**Fig. 4.13:** Experimental set up to evaluate the delay between the two PCs, equipped by Linux OS and the UDP broadcast packet sent by WLAN is used as common trigger signal between node.

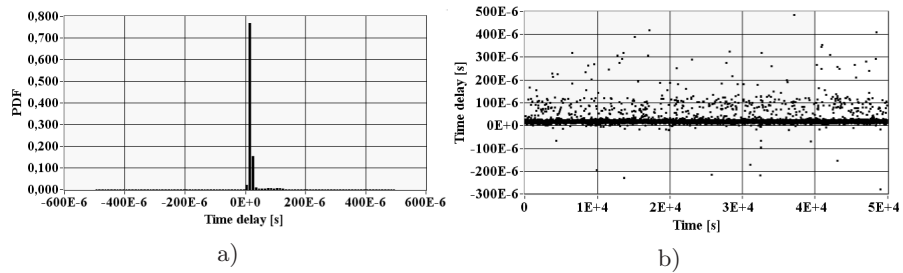
The operations performed to evaluate the delay are the same described in paragraph 4.5

Fig.4.14a) shows the PDF and Fig.4.14b) the trend versus the time of the synchronization time delay in the case both the CPU work with the same occupancy equal to 5% versus time.



**Fig. 4.14:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by WLAN and the occupancy of both the CPU nodes is 5%.

In Fig.4.15a) is shown the PDF and in Fig.4.15b) the trend versus the time of the evaluated synchronization time delay, in the case one CPU work with occupancy equal to 5%, the other one with occupancy equal to 100%.

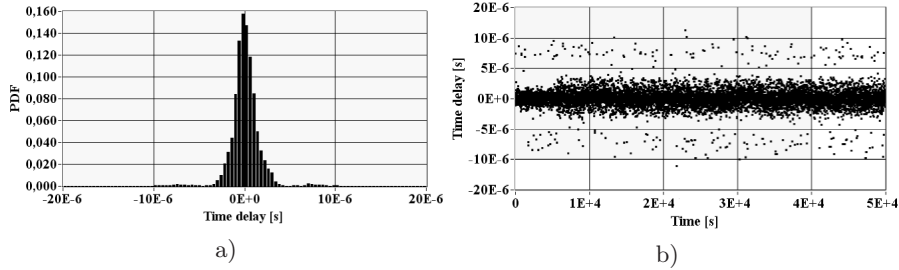


**Fig. 4.15:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by WLAN. The occupancy of one CPU is 5% and the other one is 100%.

#### 4.6.5 Wireless Lan Real Time

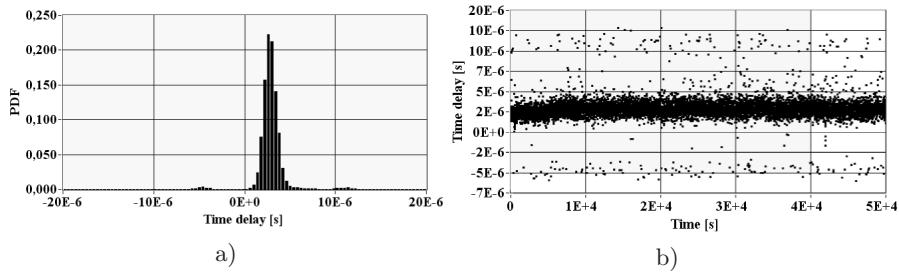
In this test, the DMS pointed out is similar to the ones used in paragraph 4.6.4, but in the Slave #1 and #2 used the Real Time OS and RTnet. Fig.4.16a)

shows the PDF and Fig.4.16b) the trend versus the time of the synchronization delay in the case both the CPU work with the same occupancy equal to 5%.



**Fig. 4.16:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by WLAN. The occupancy of one CPU is 5% and the other one is 100%.

Fig.4.17a) shows the PDF and Fig.4.17b) shows the trend versus the time of the synchronization time delay in the case one CPU work with occupancy equal to 5%, the other one with occupancy equal to 100%.



**Fig. 4.17:** a) PDF and b) trend versus the time of the synchronization time delay in the case the UDP broadcast packet is used as common trigger signal between nodes. Moreover the transmission is by RT-WLAN. The occupancy of one CPU is 5% and the other one is 100%.

Fig.4.17 highlight that even if it is present an unbalanced load on the CPUs, the trend of the delay is symmetric around 2,00 E-6 s. The causes of the change of the mean value are: (i) the unbalance load of the CPUs and the management of the priority in RTnet.

#### 4.6.6 Discussion on the experimental results

In order to furnish suggestions about the OS guaranteeing lower time delay, the experimental results, summarized in Table 4.4 are discussed.

**Table 4.4:** MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY DIFFERENT LINUX OS, CPU OCCUPANCY AND COMMUNICATION STANDARD.

	Test	Occupancy		$\mu[s]$	$\sigma[s]$
		CPU#1	CPU#2		
Lan	#1	5%	5%	1.10E-6	40.00E-9
	#2	5%	100%	14.00E-6	120.00E-9
LanRT	#3	5%	5%	56.00E-9	16.00E-9
	#4	5%	100%	6.13E-6	32.00E-9
W-Lan	#5	5%	5%	0.50E-6	40.00E-9
	#6	5%	100%	20.40E-6	200.00E-9
W-LanRT	#7	5%	5%	56.00E-9	16.00E-9
	#8	5%	100%	2.84E-6	16.00E-9

The experimental results highlight that: comparing the first and the fifth test it can be evaluated the delay introduced by the switch. Indeed, it consists only on the increasing of the mean value, due to the intrinsic stability of the hardware behavior of the switch.

All the tests confirm that the Linux OS RT operates in independent manner from the CPU occupancy, but the particular management of the priority in RTnet causes the change of the mean value. Indeed, while the process is waiting to receive the UDP packet, it loses the full control of the CPU. Therefore, if the UDP packet is coming and the CPU is busy, the process waits its turn first to retake the CPU control and to execute, in real time, the measurement procedure.

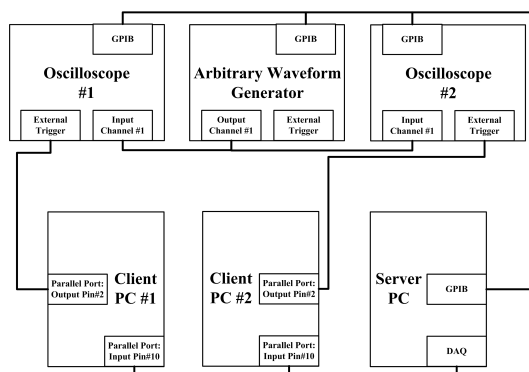
Moreover it can be assessed that by using RT OS and RTnet the mean value of the delay and the associated standard deviation are reduced in comparison With the traditional Linux OS. This occurs both in the case of Lan that in the case of Wireless Lan. Therefore, the experimental results confirm the suitability of the Real Time system to achieves synchronization in the order of  $\mu s$  also by using existing network.

#### 4.7 Validation of the deterministic OS performances by using different MI

The performance granted by the real time OS permits to investigate about the influence of the different architectures of MIs. Indeed the deterministic behavior of the OS, permits to highlight the behaviors of the different MIs.

In particular, experimental tests are executed with the aim to assess the generality of the obtained results applied in different typology of DMS. The differences consist both in the typology and characteristics of the MIs used. In all the tests the Linux OS Knoppix Real Time, Kernel 2.6 is used to equip the PC connected to the MI.

The first test refers to the set up shown in Fig.4.18.



**Fig. 4.18:** Experimental set up to evaluate the delay between the two PCs, equipped by Linux RT and connected to DSO.

The common Start Acquisition Signal (SAS) is sent to the 10th pin of the parallel port of the client PC. Then the signal is processed into the PC and the command is sent to the 2nd pin of the parallel port that is connected to the external trigger of the TDS 220 DSO. Both the oscilloscopes receive the common signal generated by the Arbitrary Waveform Generator (AWG).

The shift of the two acquired signals is the measure of the time delay caused by the PC and the MI. In particular, the procedure based on the numerical algorithm of the cross correlation is used to evaluate the time delay between the two signals.

In order to assess that the time interval multiple of the period of the common reference signal is not evaluated, the test is performed with different sampling frequency of the DSO as discussed in paragraph 3.2. Table 4.5 shows the mean value and standard deviation of the time delay by considering the TDS220 DSO.



**Table 4.5:** MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY CONSIDERING TDS220 DSO.

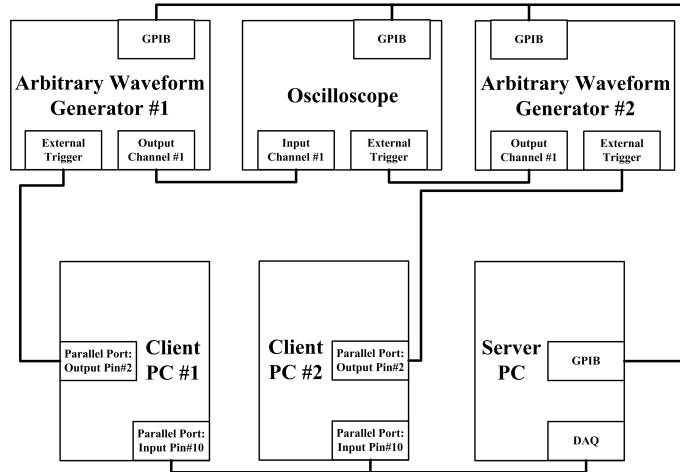
Sampling frequency [MS/s]	$\mu$ [s]	$\sigma$ [s]
10.0	4.44E-07	7.54E-09
25.0	4.51E-07	7.46E-09
50.0	4.39E-07	7.13E-09
100.0	4.43E-07	7.22E-09
125.0	4.56E-07	8.66E-09

The second test refers to the use of the TDS 7000 DSO in the set up shown in Fig.4.18. Table 4.6 shows the mean value and standard deviation of the time delay by considering the TDS7000 DSO.

**Table 4.6:** MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY CONSIDERING TDS7000 DSO.

Sampling frequency [MS/s]	$\mu$ [s]	$\sigma$ [s]
12.5	4.60E-07	7.18E-09
25.0	4.59E-07	7.42E-09
50.0	4.67E-07	7.30E-09
125.0	4.58E-07	7.16E-09

The third test refers to the set up shown in Fig.4.19. The common SAS is sent to the 10th pin of the parallel port of the client PC. Then the signal is processed into the PC and the command is sent to the 2nd pin of the parallel port, that is connected to the external trigger of the AWG2021 and AWG420. The signal of the first generator is sent to the external trigger of the oscilloscope, the signal of the second generator is sent to channel#1. The shift of the two signal is the measure of the system time delay. In order to assess that the time interval multiple of the period of the common reference signal is not evaluated, the test is performed by setting both the AWG in burst mode, with single generation. Table 4.7 shows the mean value and standard deviation of the time delay by considering the AWG instruments. Comparing the obtained results, it can be assessed that are compatible measures. Moreover, it can be also assessed that the main causes of delay are not depending on the MI of



**Fig. 4.19:** Experimental set up to evaluate the delay between the two PCs, equipped by Linux RT and connected to AWG.

**Table 4.7:** MEAN VALUE AND STANDARD DEVIATION OF THE TIME DELAY BY CONSIDERING AWG INSTRUMENTS.

Sampling frequency [MS/s]	$\mu$ [s]	$\sigma$ [s]
12.5	5.61E-07	8.76E-09
25.0	5.53E-07	8.79E-09
50.0	5.52E-07	8.62E-09
125.0	5.56E-07	8.66E-09

the node but on the interfaces. The results highlight the stability of the Linux RTAI OS [115].

## 4.8 Conclusions

In order to investigate about the effects of the non predictive OS on the communication delay, the software architectures guaranteeing the minimum time delay are analyzed.

The software solutions concern with the different setting up of non predictive Linux OS and the employment of Linux OS RT. Each solution is characterized by different lower bound of the delay and different implementing and configuring complexity.

On the basis of the results of the experimental tests, the advantages of the implementation of each of the analyzed software solutions can be evaluated. The mean value of the time delay is ranging in the interval  $[1.42 \cdot 10^{-6}s, 4.58 \cdot 10^{-5}s]$ , the standard deviation in the interval  $[0.02 \cdot 10^{-6}s, 5.92 \cdot 10^{-6}s]$ . The choice among the proposed solutions, depends on the software contest in which will be employed. In particular, the solution based on proper setting up of non predictive Linux OS is easy to implement and use, with reduced influence on the installation and the use.

Differently, the solutions based on the employment of Linux OS RT require a dept expertise for the use.

Therefore, each of the solutions is useful for the synchronization at the sub-microsecond accuracy, once the mean delay is compensated. Indeed, the standard deviation is lower than sub-microsecond. Other experimental tests has permitted to assess the independence of the achieved measurements of the delay from the architecture of the MI.



## Synchronization of MI by Using Industrial Hardware

*Abstract - It is presented the use of industrial hardware to interface the Measurement Instrument (MI) to the Distributed Measurement System (DMS). The absence of concurrent processes increases the determinism of the time execution of the process managing the MI. Nevertheless, working operation condition occurs making not deterministic the detection of the trigger condition and, consequently, making random the synchronization time delay of the MI to the node clock. In the chapter two software solutions are proposed in order to increase the synchronization accuracy achievable with the industrial hardware interface.*

### 5.1 Introduction

The research presented in this chapter is devoted to investigate about the coordination in the time domain of the node clock of the Distributed Measurement System (DMS) and the operations executed by the Measurement Instrument (MI), connected to the node by industrial hardware interface.

To avoid the random effects of concurrency of the software processes, highlighted in the Chapter 3 and 4, the interface is based on the Programmable Logic Device (PLD).

In particular the board Rabbit 4000W [108] is used. From the analysis of the operations executed by the program running on the PLD, operating conditions are shown that make not deterministic the detection of the trigger condition and, consequently, make random the variation of the synchronization time delay of the MI to the node clock. In order to detect the causes of the random variation, the polling cycle on trigger condition is taken into account and analyzed. From this analysis the model of the synchronization time delay is pointed out in order to evaluate the effects of each cause affecting the trigger check. The result of this evaluation furnishes the information to point out the adequate strategy to reduce the random variation of the detection of

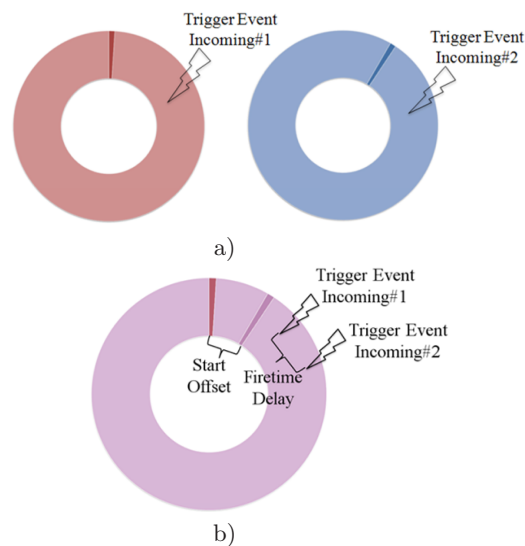
the trigger condition. Experimental tests are shown to validate the proposed strategies.

## 5.2 Interaction between polling cycle and incoming trigger

Into the DMS a synchronized node can be devoted to detect the trigger event, and then to execute the measurement procedure. The trigger can be of different nature: electromagnetic signal, TCP/UDP packets, and fire time of the synchronized clock. Therefore, the software procedure performed by the node includes the following steps:

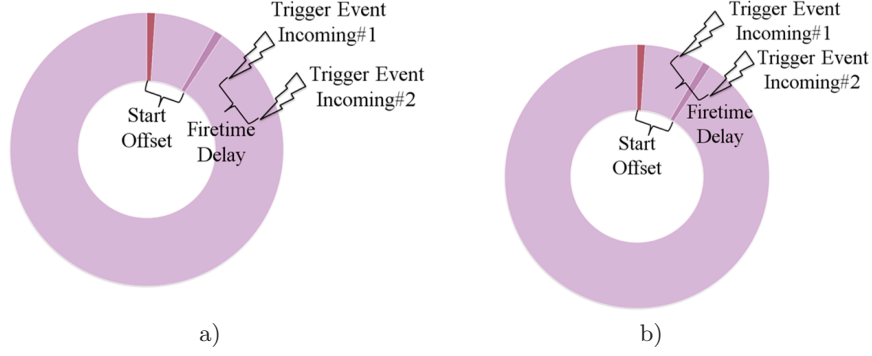
- polling on the trigger condition,
- detection of the trigger,
- transmission of the command to the MI.

Fig.5.1a) shows the polling cycle on PLD#1 and PLD#2 respectively. Fig.5.1b) shows the situation making the synchronization time delay: both the trigger conditions occur after the condition check. In this case, the trigger condition is detected at the next check after the polling cycle. This graphical representation, highlights that the synchronization time delay is depending on the behavior of both the PLDs.



**Fig. 5.1:** a) Polling cycle on PLD#1 and PLD#2, respectively; b) overlap of the previous two polling cycles.

Fig.5.2a) shows the case in which both the trigger conditions occur after the check (best case). Another best case is: the two trigger conditions occur before the check. In both this cases the synchronization time delay between the two MIs is depending only on the Start Offset, that is the time interval between the starts of the PLDs programs.



**Fig. 5.2:** a) Best case: both the triggers occur after the check; b) worst case: the trigger occurs before the check on PLD#1 and after on PLD#2.

Fig.5.2b) shows the case in which the trigger condition occurs before the check on PLD#1, and after the check on PLD#2 (worst case). Therefore, on the PLD#2 the control procedure is executed after the polling cycle.

### 5.3 Model of the synchronization time delay

In the PLD the absence of concurrency makes deterministic the execution time  $T$  of the process. It is  $T = n * f^{-1}$ , with  $n$  number of elementary operation executed,  $f$  working frequency of the PLD. Therefore, the synchronization time delay between two nodes of the DMS equipped by PLD is:

$$\Delta T = T_1 - T_2 + O_S + O_F = n \frac{1}{f_1} - n \frac{1}{f_2} + O_S + O_F = g(f_1, f_2, O_S, O_F) \quad (5.1)$$

where:

$f_1$  is the working frequency of the first PLD,  $f_2$  the working frequency of the second PLD,  $O_S$  the Start Offset,  $O_F$  the bigger multiple of  $T$  lower than the Fire Time Delay (FTD):  $O_f = \lfloor \frac{FTD}{T} \rfloor T$ . Indeed, if the FTD is lower than  $T$  the value of  $O_F$  can be considered null due to the periodicity of the check condition. In this case the effect of the FTD is taken into account considering that the trigger can be detected in one PLD first of the trigger check and after in the other one.

If the  $FTD$  is bigger than  $T$ ,  $O_F$  is equal to the higher integer value multiple of  $T$  and lower than the  $FTD$ .

On the basis of the uncertainty propagation theory [59], the variation of the synchronization time delay (5.1) is given by:

$$u_{\Delta T} = \sqrt{\left( (-nf_1^{-2})|_{\bar{f}_1, \bar{f}_2, \bar{O}_S, \bar{O}_F} u_{f_1} \right)^2 + \left( (nf_2^{-2})|_{\bar{f}_1, \bar{f}_2, \bar{O}_S, \bar{O}_F} u_{f_2} \right)^2 + u_{O_S}^2 + u_{O_F}^2} \quad (5.2)$$

In (5.2)  $u_{f_1}$  is the uncertainty of  $f_1$ ,  $u_{f_2}$  is the uncertainty of  $f_2$ ,  $u_{O_S}$  the uncertainty of the Start Offset and  $u_{O_F} = \lfloor \frac{u_{FTD}}{T} \rfloor T$ , where  $u_{FTD}$  is the uncertainty of FTD.

The (5.2) is valid in absence of "conditional jump" and correlation between parameters. This modifies the number of operations to be executed on the basis of the particular event.

The time delay introduced by each PLD, is:

$$T_i = \begin{cases} n_{1a} \frac{1}{f_1} + n_2 \frac{1}{f_1} & \text{best case} \\ n_{1b} \frac{1}{f_1} + n_2 \frac{1}{f_1} & \text{worst case} \end{cases} \quad (5.3)$$

where  $n_{1a}$  is the number of operation to detect the trigger condition,  $n_2$  is the number of operation to transmit the command to the MI,  $n_{1b}$  is equal to  $n_{1s} + n_p$ , with  $n_p$  is the number of operations of the polling cycle.

The PLD detects the trigger condition only after the check of the condition (best case). If the trigger occurs after the check it will be detected at the next check, then after the polling cycle (worst case).

The trigger event can be detected from one PLD before another one on the basis of the following conditions:

1. the polling cycle is completed or started before another one;
2. the trigger signal, in the case of electromagnetic signal, propagates before another one;
3. the fire time of the trigger at one synchronized node occur earlier of another one as a consequence of the uncertainty of the synchronization method used to synchronize the clock nodes of DMS.

In the case the two trigger conditions are detected in the same polling cycle, the best case occurs. Therefore, by posing  $n_{1b} = n_{1a} + n_p$ , the synchronization time delay is:



$$\Delta T = \begin{cases} n_{1a} \frac{1}{f_1} + n_2 \frac{1}{f_1} - n_{1a} \frac{1}{f_2} - n_2 \frac{1}{f_2} + O_S + O_F & (a) \\ (n_{1a} + n_p) \frac{1}{f_1} + n_2 \frac{1}{f_1} - (n_{1a} + n_p) \frac{1}{f_2} + O_S + O_F & (b) \end{cases} \quad (5.4)$$

In particular the 5.4(a) refers to the best case occurring in both the PLDs, the 5.4(b) refers to the worst case occurring in only one PLD.

In the case the two trigger conditions are detected with delay of one polling cycle, the worst case occurs. It is to highlight that the term  $O_F$  does not depend on the PLD but on the particular synchronization techniques taken into account. Therefore, the synchronization time delay is:

$$\Delta T = \begin{cases} (n_{1a} + n_p) \frac{1}{f_1} + n_2 \frac{1}{f_1} - n_{1a} \frac{1}{f_2} - n_2 \frac{1}{f_2} + O_S + O_F & (a) \\ n_{1a} \frac{1}{f_1} + n_2 \frac{1}{f_1} - (n_{1a} + n_p) \frac{1}{f_2} - n_2 \frac{1}{f_2} + O_S + O_F & (b) \end{cases} \quad (5.5)$$

In particular the 5.5(a) refers to the worst case occurring in the first PLD, the 5.5(b) refers to the worst case occurring in the second PLD.

The variation of the synchronization time delay from (5.4) is:

$$u_{\Delta T} = \begin{cases} \sqrt{\left( -(n_{1a} + n_2) f_1^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_1} \right)^2 + \left( +(n_{1a} + n_2) f_2^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_2} \right)^2 + u_{O_S}^2 + u_{O_F}^2} & (a) \\ \sqrt{\left( -(n_{1a} + n_p + n_2) f_1^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_1} \right)^2 + \left( +(n_{1a} + n_p + n_2) f_2^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_2} \right)^2 + u_{O_S}^2 + u_{O_F}^2} & (b) \end{cases} \quad (5.6)$$

In particular the 5.6(a) refers to the best case occurring in both PLDs, the 5.6(b) refers to the worst case occurring in both PLDs.

The variation of the synchronization time delay from (5.5) is:

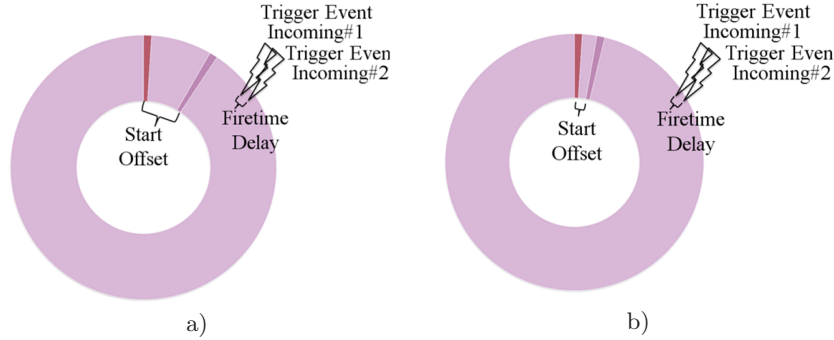
$$u_{\Delta T} = \begin{cases} \sqrt{\left( -(n_{1a} + n_p + n_2) f_1^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_1} \right)^2 + \left( +(n_{1a} + n_2) f_2^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_2} \right)^2 + u_{O_S}^2 + u_{O_F}^2} & (a) \\ \sqrt{\left( -(n_{1a} + n_2) f_1^{-2} \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_1} \right)^2 + \left( +(n_{1a} + n_p + n_2) \Big|_{\bar{f}_1, \bar{f}_2, \bar{O}} u_{f_2} \right)^2 + u_{O_S}^2 + u_{O_F}^2} & (b) \end{cases} \quad (5.7)$$

In particular the 5.7(a) refers to the worst case occurring in the first PLD, the 5.7(b) refers to the worst case occurring in the second PLD.

The (5.5) and (5.7) show that the synchronization time delay and the variation of the synchronization time delay can be reduced by reducing  $n_p$  as much as possible. This solution has been experimental implemented by using the PLD equipped by modified software driver of the WiFi interface [55].

In conjunction to the previous one, others possibilities available to reduce the variation of the synchronization time delay are the following:

- reduction of the uncertainty of the incoming trigger. In this case the probability that the worst case occurs can be reduced (Fig.5.3a). This reduction can be achieved by using synchronization protocol of the clock nodes of the DMS guarantying the lower uncertainty. For example, the use of the standard IEEE 1588 grants a sub microsecond accuracy, in the contrary, the NTP protocol [93] guarantees the millisecond accuracy;
- reduction of the Start Offset, as shown in Fig.5.3b);
- decrease of the ratio between the Start Offset and the execution time of the polling cycle.



**Fig. 5.3:** a) Uncertainty reduction of the incoming trigger can reduce the probability that the worst case, occurs, b) the reduction of the Start Offset can reduce the probability that the worst case, occurs.

### 5.4 Software solutions reducing the execution time of the polling cycle

The working operations performed by the board Rabbit RCM4400W on the incoming wireless packet are the following. The electromagnetic wave at the antenna is converted in bit streams by the electronic circuit. Successively, the bit stream is converted according to the MAC level of the ISO/OSI standard by the FPGA and stored into the register inside the FPGA. Fig.5.4 shows the block scheme of the normal operations executed by the control program running into the board.

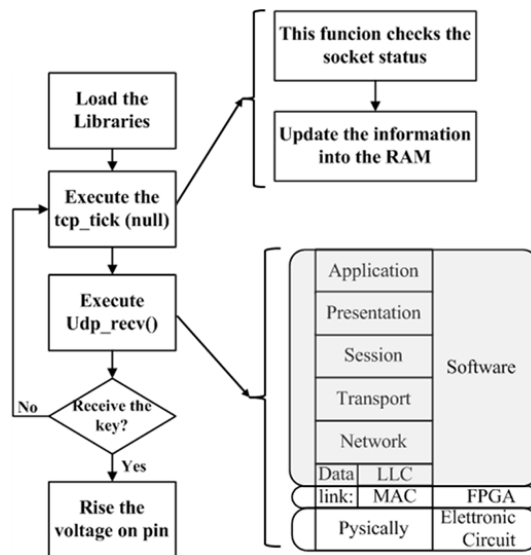


Fig. 5.4: Block scheme of the operations executed by the original control program running into the board Rabbit RCM4400W.

When the program checks the incoming packet from the network, it invokes the WiFi driver. The instruction `Tcp_tick(null)` drives the "background" processes to update information and tests the socket [32].

The WiFi driver executes all the protocol stack ISO/OSI and returns the data to the control program. With the instruction `UDP_recv()`, block scheme of Fig.5.4, the control program requires the data. This instruction is not blocking. In particular, if the data is not ready into the FPGA register, all the operations will be executed on inconsistent data, and the polling cycle is executed to wait the incoming UDP packet. Fig.5.5 shows the operations performed into the polling cycle.

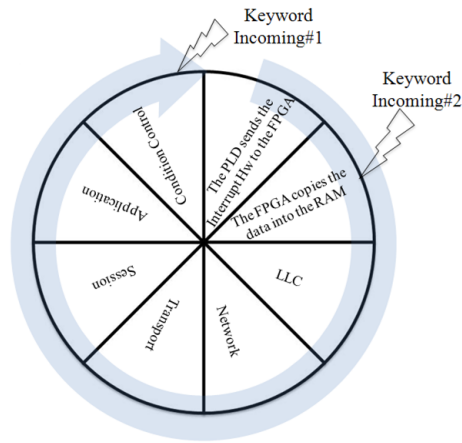


Fig. 5.5: Operations performed into the polling cycle.

In order to reduce the execution time of the polling cycle, the proposal concerns with: (i) the reduction of the number of operations executed when the data is still not received, and (ii) the reduction of the stack to check the trigger condition.

As concerning the first proposal, as shown in the modified block scheme of Fig.5.6, after the updating of the received data in the RAM, it is checked if the number of the incoming packet is not zero.

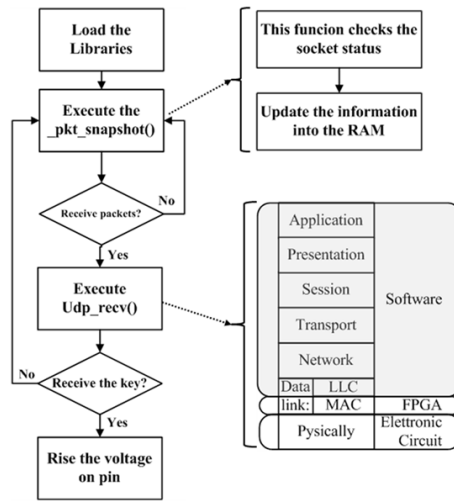
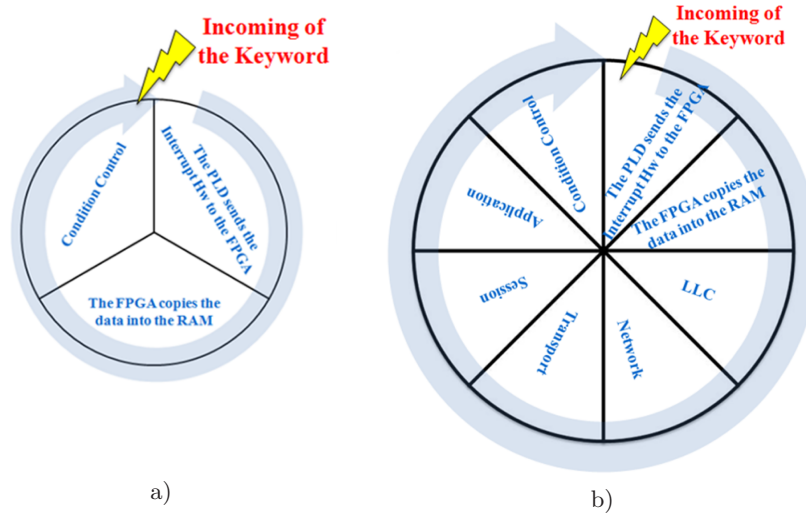


Fig. 5.6: Block scheme of the operations executed by the modified control program running into the board Rabbit RCM4400W in order to reduce the time of the polling cycle.

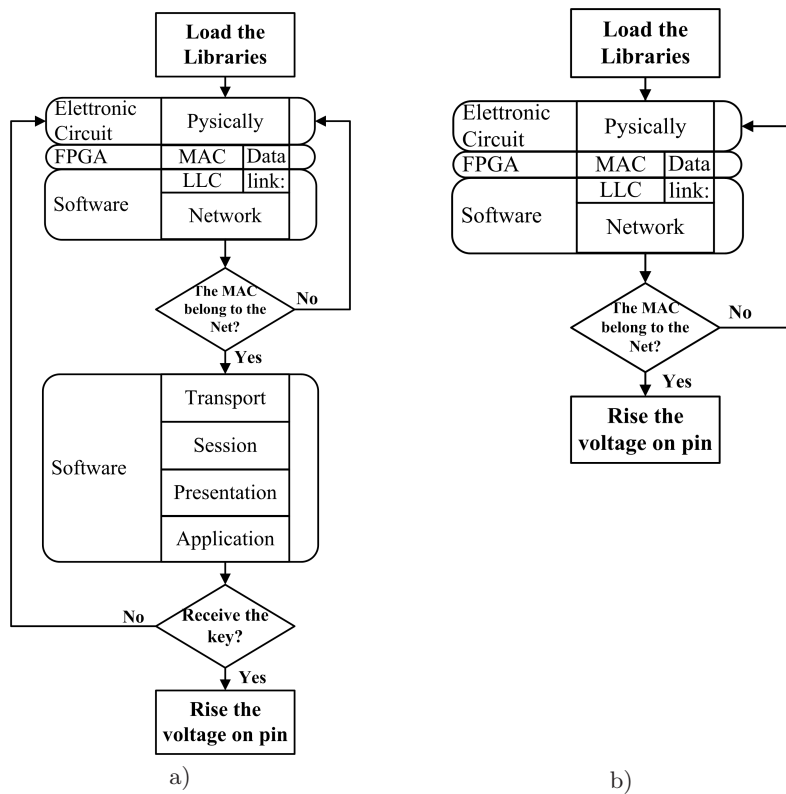
Therefore, on the basis of the mathematical model presented in the third paragraph, if the worst case occurs, the delay introduced by the missing check is strongly reduced as shown in Fig.5.7.



**Fig. 5.7:** Operation executed when a) the worst case occurs, b) the best case occurs.

As concerning with the second proposal, the check of the MAC address of the sender is used to determine if the data are consistent for the application. Indeed, in the wireless network the problem of sharing the physical device is common and access policies avoiding the contemporaneous use of the physical device are present in literature. Nevertheless, there are no mechanism to assess the packets reception from only the sender involved in the communication. The packet received from outsider the network under consideration has not semantic consistence. The first proposal cannot distinguish it because takes into account the syntactic consistence. The check on the sender identity is a significant improvement in order to reduce the execution time. The MAC address is univocal address assigned at each network adapter and implemented usually in the hardware.

The performed research is addressed to control the MAC address directly at network level. The block scheme of Fig.5.8a) shows the operations performed at the receiver in normal condition. If the sender signal is used only for the wireless trigger, it is enough to insert into the driver the routine to rise the voltage level on the pin connected to the MI trigger, as shown in Fig.5.8b).

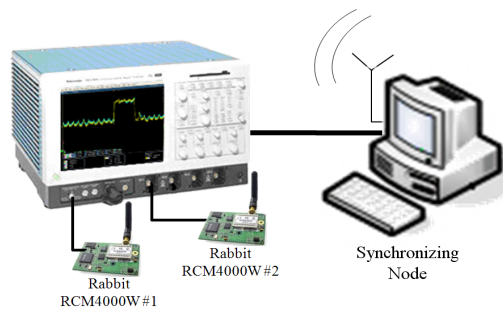


**Fig. 5.8:** a) Block scheme of the operations executed by the control program using the modified software driver. b) Block scheme of the operations executed by the control program using the modified software driver in the case the sender signal is used only as wireless Trigger.

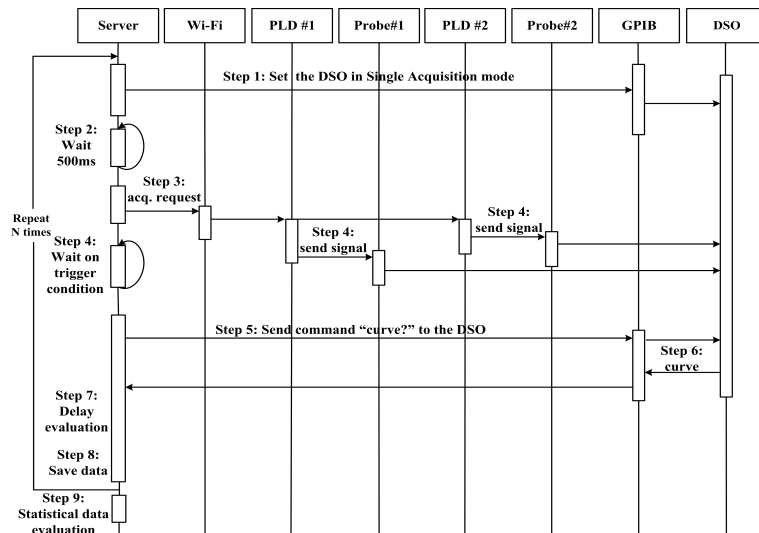
### 5.5 Experimental tests

The experimental tests are performed to validate the previous solutions pointed out regarding the reduction of the execution time of the polling cycle. The considered DMS is constituted by 3 nodes. The master node is equipped by PC sending the command to start acquisition to the two slave nodes. Each of the two slave nodes is equipped by Digital Storage Oscilloscope (DSO) Tektronix TDS7404 interfaced to the DMS by the board Rabbit RCM4400W. This board is equipped by the PLD RCM4400W, and WiFi adapter [108].

In Fig.5.9 is proposed the DMS used to perform the tests. The sequence of the commands executed are reported in the block diagram of Fig.5.10.



**Fig. 5.9:** DMS to evaluate the synchronization time delay introduced by the Rabbit RCM4000W.



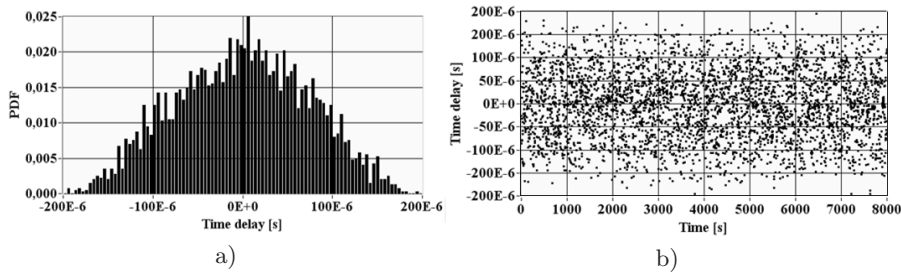
**Fig. 5.10:** Sequence of the commands planned to evaluate the delay introduced by the PLDs.

The delay is evaluated by the sample number occurring between the trigger and the cross of the established threshold by the signal acquired. Once set the trigger at 50% of the memory length, it needs to detect the first sample overcoming the threshold, only. In particular, denoted by  $i$  the half of the record length,  $j$  the index corresponding to the sample overcoming the threshold,  $dt$  the time between two samples, the time delay between the trigger and the signal is:

$$\Delta t = |(i - j) * dt| \tag{5.8}$$

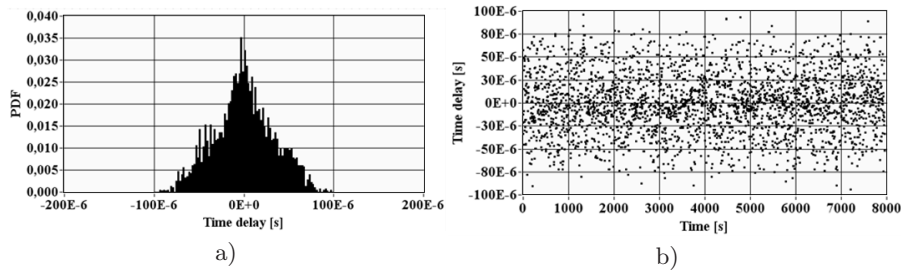
Experimental test are performed to evaluate a) the Probability Density Function (PDF) and b) the trend versus the time of the delay values in the following cases:

1. original program, block scheme of Fig.5.4, implemented by using the software driver furnished with the board Rabbit 4000W (Fig.5.11a) and b));



**Fig. 5.11:** a) Trend versus the acquisition time, and b) PDF of the synchronization time delay obtained applying the original program.

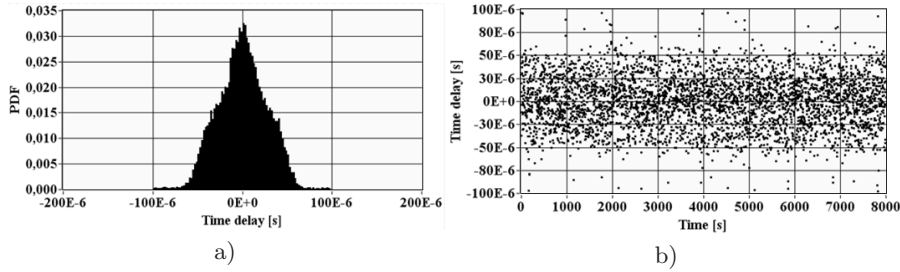
2. modified program execution, block scheme of Fig.5.6, in which the procedure to check the size of the incoming data is imported directly in the software implemented to drive the MI (Fig.5.12a) and b));



**Fig. 5.12:** a) Trend versus the acquisition time, and b) PDF of the synchronization time delay obtained applying the modified program.



- modified software driver, block scheme of Fig.5.8b), in which the driver furnished with the board Rabbit 4000W is modified to distinguish the case of packet not arrived or coming from a device does not belonging to the DMS (Fig.5.13a) and b)).



**Fig. 5.13:** a) Trend versus the acquisition time, and b) PDF of the synchronization time delay obtained applying the modified software driver.

The PDF is evaluate on the record of 4000 samples acquired each 2s. The results of the experimental tests are summarized in Table 5.1.

**Table 5.1:** EVALUATION OF THE SYNCHRONIZATION TIME DELAY BY USING THE ORIGINAL CONTROL PROGRAM OF THE BOARD RABBIT RCM4400W AND THE MODIFIED ONE.

Control program	Synchronization time delay	
	$\mu$ [s]	variation [s]
original program	748.77E-9	73.07E-6
modified program	120.00E-9	33.18E-6
modified software driver	14.28E-9	27.33E-6

The experimental tests validate the proposed strategy.

## 5.6 Conclusions

The analysis of the operations executed by the control program running on the PLD has shown that operating conditions can occur making not deterministic the detection of the trigger condition and, consequently, making random the synchronization time delay of the MI to the node clock.

On the basis of this analysis the model of the variation affecting the synchronization time delay is pointed out. Owing to this model the effects of

each cause affecting the trigger check are evaluated. Consequently, adequate strategy to reduce the random variation of the detection of the trigger condition is pointed out. The proposed strategy concerns with the reduction of: (i) the time interval of the polling cycle, (ii) the uncertainty of the incoming trigger, (iii) the Start Offset and the ratio between the Start Offset and (iv) the execution time of polling cycle.

The experimental tests confirm the validity of the mathematical model and the effectiveness of the proposed software solution.

## Synchronization of MI by Using Embedded Hardware

*Abstract - A new architecture of Hardware Interface (HI) is proposed to execute the trigger command for the Measurement Instruments cooperating in the Distributed Measurement System. The new HI (i) avoids the random effects of concurrency of the software processes running on PC, (ii) reduces the upper bound of the execution time of the procedure based on Real Time Operating System, and (iii) reduces the random causes on the time delay, introduced by the polling cycle, to detect the trigger condition on the Programmable Logic Device (PLD).*

### 6.1 Introduction

In the fifth chapter was proposed the solution based on the use of the Programmable Logic Device (PLD) to interface the Measurement Instrument (MI) to the Distributed Measurement System (DMS). The advantage is that the concurrency causes among the software processes are completely avoid, and, consequently, the execution time of the interesting procedure devoted to the command execution for the MI should be deterministic.

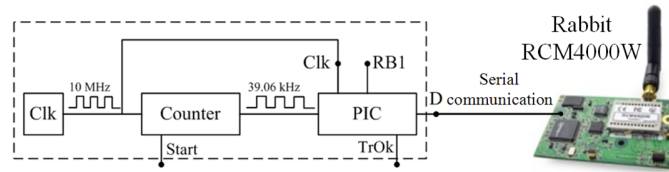
Nevertheless, operating conditions can occur making not deterministic the detection of the trigger condition to start the command execution for the MI. If the trigger occur after the check it will be detected at the next check. In this case, the final result is that the start time of the command execution is random and depends on the time delay between the trigger and the trigger check.

In order to overcome the inconvenient arising from the random time delay of the start time of the MI [55], caused by the polling cycle, in the chapter a different Hardware Interface (HI) is taken into account to interface the MI to the node of the DMS. The proposed architecture of the HI includes the PLD as wireless interface, and the board equipped by PIC, Counter block, and Clk block. The board is designed to reduce the random variation of the time delay and to avoid the polling cycle to check the trigger condition. From

the analysis of the HI the models of the synchronization time delay and its variation are pointed out in order to evaluate the effects of each cause affecting the execution of the trigger command for the MI. This evaluation furnishes the information to point out the adequate strategy to reduce the random variation of the time delay to synchronize the MIs. Experimental tests validate the HI pointed out and the proposed strategy.

## 6.2 Performed operation by proposed HI

Fig.6.1 shows the hardware architecture of the new proposed HI connecting the MI to the node of the DMS. It is constituted by: (i) PIC16F84a, (ii) 8-bit Counter block, (iii) Clk block equipped by 10MHz temperature compensated crystal oscillator, and (iv) board Rabbit RCM4400W [108].

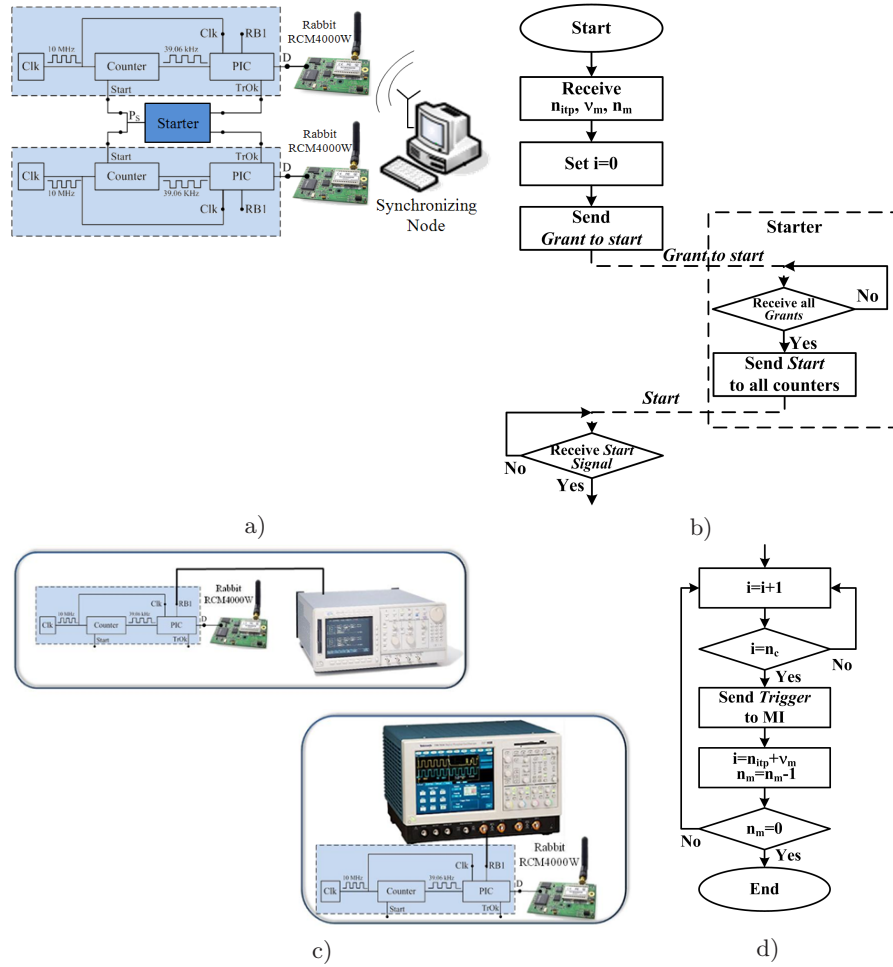


**Fig. 6.1:** Block scheme of the proposed HI architecture reducing the random time delay variation to check the trigger condition.

The board Rabbit RCM4400W receives from the Master PC on the WiFi interface the information about the trigger condition to execute the command for the MI, and transmits it on the serial interface, pin D, to the PIC [8]. The software running on the PIC checks the trigger condition by means of the interrupts arising from the Counter block, and sends the command to the MI by rising the signal on the pin RB1. The Clk block is used as reference clock to temporize with high accuracy the command execution from the PIC and the Counter block.

The Counter block is used as frequency divider. Indeed, because the Clk block has frequency equal to 10MHz and the number of bit of the Counter block is 8bit, the PIC receives the interrupt each  $25,6\mu s$ , while the elementary operations, required to check the trigger condition, are performed by the PIC in  $12,5\mu s$ . In this way the end of performed operations is guaranteed before the successive interrupt occurs.

The pins TrOk and Start are used to perform the synchronization of the Counter block of two different HIs by means of the external Starter block. This last operates by synchronizing the start of the two Counter blocks, as shown in Fig.6.2 a) and Fig.6.2 b).



**Fig. 6.2:** a) Block scheme of the connection into the Synchronization Phase, and b) block diagram of the performed operation by HI into the Synchronization Phase; c) block scheme of the connection into the Operative Phase for the synchronization of two stand alone MIs, and d) block diagram of the performed operation by HI into the Operative Phase.

The Starter block is constituted by two latches and only one port and. While each input does not receive the high signal level from the pin TrOk of the two HIs, the signal at the output pin Ps is low. Once received all the agreements, the output signal at the pin Ps is high. This signal is also used to reset the Starter latches.

On each HI, a latch is connected to the pin Start of the Counter block to ensure the correct functioning once the Starter is disconnected.

### 6.3 Synchronization by means of proposed HI

Fig.6.2 shows the block scheme of the hardware connections and the block diagram of the performed operations in (i) the First Phase in which the synchronization is performed between two HIs, Fig.6.2 a) and b), and (ii) the Second Phase in which the HIs are employed to synchronize two stand alone MIs [54], Fig.6.2 c) and d).

The First Phase, Fig.6.2 a), involves the Master PC to send the trigger condition at each HI, and the external Starter block to synchronize the start of the counting between two HIs. The trigger condition corresponds to the number  $n_c = n_{itp}2^N$ , with  $n_{itp}$  number of interrupts that the PIC must count before to execute the command for the MI, and  $N$  bit number of the Counter block. The Master PC evaluates  $n_{itp}$  according to the frequency of the Clk block and the bit number of the Counter block. The Master PC sends, also, the information concerning with the number of repetitive measures  $n_m$  with period  $\nu_m$ . In the case only one measure is requested, is  $n_m = 1$ . Once set into the PIC the internal variable  $n_{itp}$ ,  $n_m$ ,  $\nu_m$ , the HI is ready to start the time counting. Therefore, it sends to the Starter block the *grant to start* signal by rising the voltage level on the pin TrOk, Fig.6.2 b).

Once received all the grants, the Starter block sends to the two Counter the *Start to count* signal, by rising the voltage on pin Ps. Because each HI receives the signal in the parallel modality, the time delay between the gate opening of the two Counter blocks can be assumed negligible. Consequently, the interrupt sent by each Counter block is synchronized with the other ones, and, as a consequence, the corresponding PICs are forced to operate in synchronized modality.

Once the Counter block start the count, the Second Phases begins, Fig.6.2 c). Each HI is: (i) disconnected from the external Starter block, (ii) moved to the measurement place, and (iii) connected to the MI. During this operation mode, each PIC receives the interrupt from the Counter block in synchronized modality with other PICs. The operations performed by the PIC at each interrupt are the following, Fig.6.2d): (i) the internal variable  $i$  is incremented, and (ii)  $i$  equal to  $n_{itp}$  is checked. When this condition occurs, HI sends the trigger signal to the MI by the pin RB1, and the new values are upgraded  $n_{itp} = n_{itp} + \nu_m$ ,  $n_m = n_m - 1$ . If the condition  $n_m = 0$  occurs, the measurement procedure is considered done and the PIC is waiting for new trigger condition from the Master PC.

### 6.4 Model of random variation of the synchronization time delay by using the proposed HI

The synchronization time delay  $T$  between two stand alone MIs, each one equipped by the HI shown in Fig.6.1, is:

$$\Delta T = \left( n_{c1} \frac{1}{f_c} - n_{c2} \frac{1}{f_{c2}} \right) + \left( n_{p1} \frac{1}{f_{p1}} - n_{p2} \frac{1}{f_{p2}} \right) + O_C + O_P \quad (6.1)$$

where  $f_{c1}$  is the frequency of the oscillator of the HI#1 equipping the MI#1 and  $f_{c2}$  that of the oscillator of the HI#2 equipping the MI#2;  $f_{p1}$  and  $f_{p2}$  the oscillator frequency of the PIC#1 and PIC#2, respectively;  $n_{c1}$  and  $n_{c2}$  the counting number of HI#1 and HI#2, respectively;  $n_{p1}$  and  $n_{p2}$  the number of elementary operations needs to execute the procedures on the PIC#1 and PIC#2, respectively;  $O_C$  the start offset time between the Counter on the HI#1 and that on HI#2, and  $O_P$  the start offset time between the two PICs.

The start offset  $O_P$  can be neglected because the following condition occurs: the storage of the internal variable  $n_c$ ,  $n_m$ ,  $\nu_m$  into each PIC ends before the incoming of the first interrupt signal generated by the associated Counter. Indeed, the external Starter block sends the *start to count* signal to the two Counters once received the *grant to start* signals by both the PICs, Fig.6.2 b).

Under the condition that the PIC performs the operations to check the trigger condition before of the successive incoming interrupt, it is possible to assess that the delay to perform the operations in response to the interrupt  $n$  is not cumulative with the one cumulated to perform the operation corresponding to the interrupt  $n + 1$ . Therefore, the operations taken into account in  $n_{p1}$  and  $n_{p2}$ , respectively, are the following: (i) the internal counter is incremented, (ii) the occurrence of the trigger condition is checked, and (iii) the trigger procedure is executed.

Therefore the (6.1) can be rewritten as:

$$\Delta T = \left( n_{c1} \frac{1}{f_{c1}} - n_{c2} \frac{1}{f_{c2}} \right) + \left( n_{p1} \frac{1}{f_{p1}} - n_{p2} \frac{1}{f_{p2}} \right) + O_C \quad (6.2)$$

In order to evaluate the  $\Delta T$  variation it can be noted that this is the same case of the evaluation of the uncertainty in the case the measurand  $Y$  is determined from  $N$  other quantities  $X_1, X_2, \dots, X_N$  through a functional relationship [57]:

$$T : Y = f(X_1, X_2, \dots, X_N) \quad (6.3)$$

Therefore, the law of propagation of uncertainty can be applied in order to establish the functional relationship among the  $\Delta T$  variation and the variation of the above mentioned quantities [58]. In particular, because it is:

$$f : \Delta T = f(n_{c1}, f_{c1}, n_{c2}, f_{c2}, O_C, n_{p1}, f_{p1}, n_{p2}, f_{p2}) \quad (6.4)$$

the variation of the synchronization time delay is:

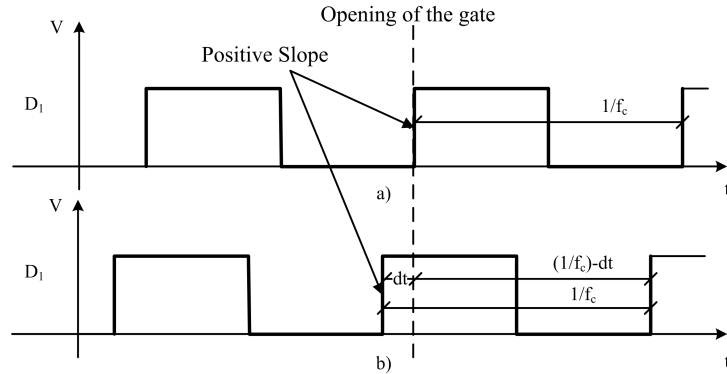
$$u_{\Delta T} = \sqrt{\begin{aligned} &((-n_{c1}f_{c1}^{-2})|_{\bar{x}} u_{f_{c1}})^2 + (n_{c2}f_{c2}^{-2}|_{\bar{x}} u_{f_{c2}})^2 + u_{O_C}^2 + \\ &+ (n_{p2}f_{p2}^{-2}|_{\bar{x}} u_{f_{c2}})^2 + (-n_{p1}f_{p1}^{-2}|_{\bar{x}} u_{f_{c2}})^2 \end{aligned}} \quad (6.5)$$

where  $u_{f_{c1}}, u_{f_{c2}}$ , are the variation of  $f_{c1}, f_{c2}$ , respectively,  $u_{f_{p1}}, u_{f_{p2}}$ , variation of  $f_{p1}, f_{p2}$ , respectively, and  $u_{O_c}$  the variation of  $O_C$ .

The (6.5) is valid in both absence of conditional jump and correlation between parameters. Indeed, the conditional jump modifies the number of operations,  $n_{p1}$  and  $n_{p2}$ , to be executed on the basis of the particular event.

In the case under examination, it is assumed  $f_{c1} = f_{p1}$ , and  $f_{c2} = f_{p2}$ , according to the block scheme of Fig.6.1. Moreover, for synchronization purpose can be imposed  $f_{c1} = f_{c2} = f_c$ , and  $n_{c1} = n_{c2} = n_c$ . Because the PIC#1 and PIC#2 execute the same procedure, it is  $n_{p1} = n_{p2} = n_p$ . The offset  $O_C$  depends on the offset between the signals fed by the two Clk blocks used as input to the two Counter blocks at the start of the count. If these two signals are not synchronized, and the Counter block is sensible only to the positive slopes of the input signal,  $O_c$  is the time delay between their positive slopes.

Therefore, considering negligible the delay between the time opening of the doors of the counters in HI#1 and HI#2,  $O_c = D_1 - D_2$ , where  $D_1$  is the delay between the start of count and the incoming of the first positive slope of the signal incoming from Clk on HI#2,  $D_2$  is the delay between the start of count and the incoming of the first positive slope of the signal incoming from Clk on HI#1. In the follow it is assumed that the value of  $D_1$  and  $D_2$  is distributed uniformly in the range  $[0 - f_c^{-1}]$  as shown in Fig.6.3.



**Fig. 6.3:** Delay between the opening of the gate in the counter and the effective start to count. In a) the best case occurs, and the delay is zero. In b) the middle case occurs and the delay is equal to  $\frac{1}{f_c} - dt$ .



Indeed, taking into account only one signals, the delay  $D_1$  can be equal to zero, Fig.6.3a) or distributed uniformly between 0 and  $f_c^{-1}$ , as shown in Fig.6.3b).

Fig.6.3a) shows the case in which the positive slope of the signal occurs just before the gate is opening. Fig.6.3b) shows the case in which the positive slope of the signal occurs after  $dt$  the gate is open. Therefore, the counter will count the first positive slope after  $f_c^{-1} - dt$ . The worst case is for  $dt \rightarrow 0^+$ .

Thus,  $D_1$  and  $D_2$  are characterized by mean value equal to  $\frac{f_c^{-1}}{2}$  and standard deviation equal to  $\frac{f_c^{-1}}{2}$ . Therefore according to the Guide to Uncertainty Measure [59] it is:

$$u_{O_C} = \sqrt{\left(\frac{\partial O_C}{\partial D_1}\right)_{D_1, D_2}^2 u_{D_1}^2 + \left(\frac{\partial O_C}{\partial D_2}\right)_{D_1, D_2}^2 u_{D_2}^2} \quad (6.6)$$

Because  $\left(\frac{\partial O_C}{\partial D_1}\right)_{D_1, D_2} = \left(\frac{\partial O_C}{\partial D_2}\right)_{D_1, D_2} = 1$  the (6.6) can be rewritten as:

$$u_{O_C} = \sqrt{u_{D_1}^2 + u_{D_2}^2} \quad (6.7)$$

Because  $u_{D_1}$  and  $u_{D_2}$  are evaluated on the basis of the uniform distribution, this case is similar to the case of uncertainty of type B. Therefore, in order to be used in the (6.7), they must be divided by  $\sqrt{3}$  [59]. Substituting the values of  $u_{D_1}$  and  $u_{D_2}$  in (6.7) and reducing the equation in its simplest form it is:

$$u_{O_C} = \sqrt{2 \left(\frac{1}{2\sqrt{3}f_c}\right)^2} = \frac{1}{\sqrt{6}f_c} \quad (6.8)$$

Under the previous conditions, the (6.5) can be rewritten as:

$$u_{\Delta T} = \sqrt{\left((-n_{c1}f_{c1}^2)_{\bar{x}} u_{f_1}\right)^2 + \left(n_{c2}f_{c2}^{-2} u_{f_{c2}}\right)^2 + u_{O_C}^2 + \left(n_{p2}f_{p2}^{-2} u_{f_{p2}}\right)^2 + \left(-n_{p1}f_{p1}^2 u_{f_{p2}}\right)^2} \quad (6.9)$$

Then:

$$u_{\Delta T} = \sqrt{\frac{12f_c^{-2}u_{f_c}^2(n_c^2 + n_p^2) + 1}{6f_c^2}} \quad (6.10)$$

The (6.10) suggests that to reduce the variation of the synchronization time delay, it is necessary to increase the clock frequency and its accuracy.

### 6.5 Actual realization of the proposed HI

In order to execute the experimental tests, two HI are built. In Fig.6.4 is shown the block scheme of the HI, in Fig.6.5 is shown the block scheme of the HI and the Starter Block.

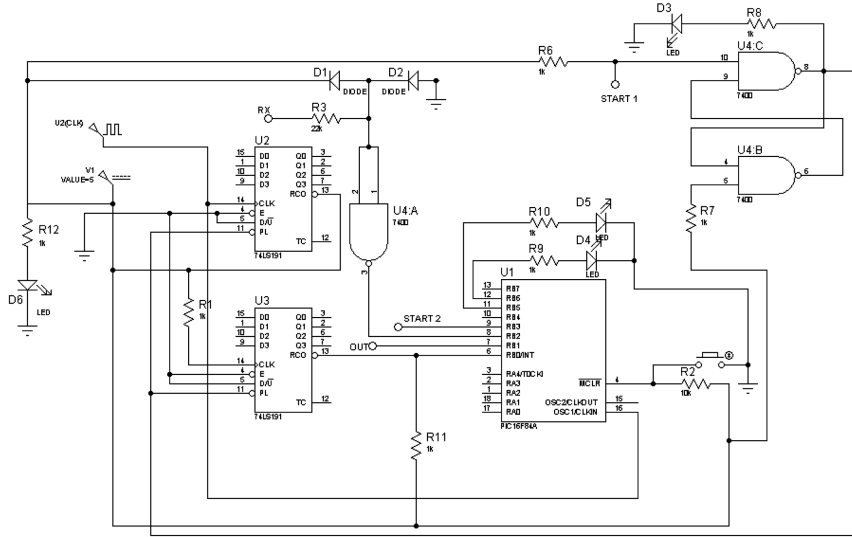


Fig. 6.4: Block scheme of the HI.

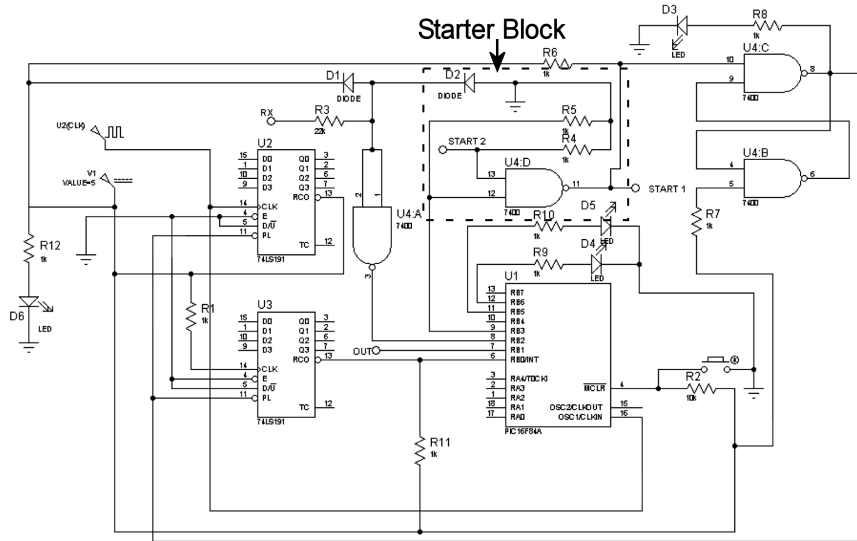


Fig. 6.5: Block scheme of the HI and of the Starter Block.

For sake of completeness, Table 6.1 shows all the used components and their characteristics.

**Table 6.1:** LIST OF THE COMPONENTS USED TO BUILD THE HI ACCORDING TO THE BLOCK SCHEME OF FIG. 6.4 AND FIG. 6.5.

Resistor	R1=R2=(10.0 ± 0.5)kΩ
	R3=(22.0 ± 1.2)kΩ
	R4-R12= (1.00 ± 0.05)kΩ
Diode	D1-D2=1N4001
	D3-D6=LED
Integrated circuit	U1=PIC16F84A
	U2-U3=74LS191
	U4=74HC00
Capacitor	C1=C2= (22.00 ± 0.44)pF

## 6.6 Experimental tests

Experimental tests are executed in order to validate (i) the correctness of the performed operations by the proposed HI, (ii) the model of the synchronization time delay (6.2), and (iii) the model of the variation of the synchronization time delay (6.10).

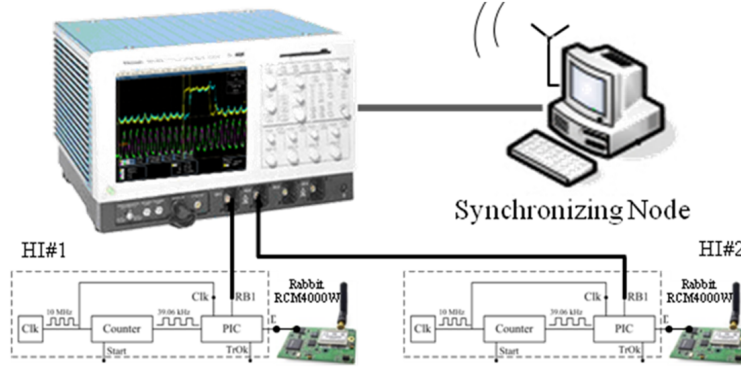
Beginning, the two HIs are synchronized according to the block scheme of the connection into the Synchronization Phase shown in Fig. 6.2a).

The block diagram of Fig. 6.2 b) shows the performed operation into the Synchronization Phase. Successively, the two HIs are disconnected from the external Starter block, moved to the measurement place, and made available for the Operative Phase according to the diagram of the performed operation shown in Fig. 6.2d).

Because the intent is to detect and measure the time delay between the trigger signals furnished at the pin RB1 of the HI#1 and HI#2, each one is connected to the input channel of the DSO, as shown in Fig. 6.6.

The Master PC is connected by GPIB interface bus to the DSO, and sends the trigger condition to the two HIs by WiFi connection. In the experimental tests the trigger conditions are the same for both the HIs. In particular, they are:  $n_{c1} = n_{c2} = 10^7$ ,  $n_m = 1$ ,  $\nu_m = 0$ ,  $n_{p1} = n_{p2} = 126$ . The trigger signal fed from the HI#1 is sent to the Ch#1 of the DSO, that of the HI#2 is sent to the Ch#2 of the DSO.

The time delay between these two trigger signals furnishes the evaluation of the delay caused by the hardware and the software of the two HIs.



**Fig. 6.6:** Experimental set up to evaluate the delay between the trigger signals generated by two HIs.

The delay is evaluated by means of the number of samples occurring between the cross of the established threshold by the two input signals. In particular, denoted by  $i$  and  $j$  the indexes corresponding to the samples overcoming the threshold,  $dt$  the time between two samples, the time delay is:

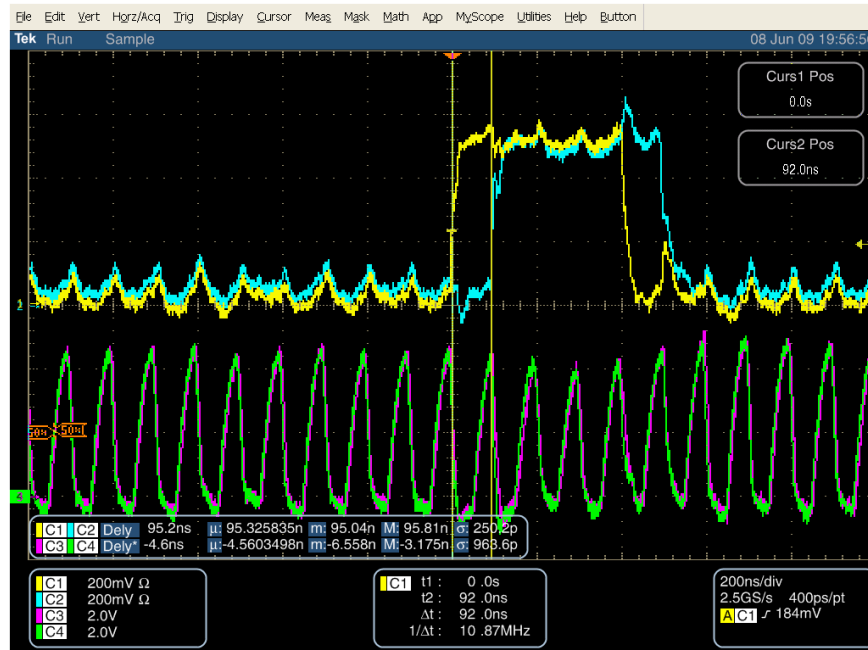
$$\Delta t = |(i - j) * dt| \quad (6.11)$$

Fig.6.7 shows the snapshot of the DSO to measure the synchronization time delay between the two trigger signals. The measured time delay is  $\Delta T = 92ns$ .

The Ch#3 and Ch#4 show the output signals of the Clk blocks of HI#1 at the frequency  $f_{c1}$  and of HI#2 at the frequency  $f_{c2}$ , respectively. In the case of Fig.6.7, the differences between the mean values and standard deviation of  $f_{c1}$  and  $f_{c2}$  can be assumed negligible. This experimental result validates the assumption that  $f_{c1} = f_{c2}$ . The common value is  $f_c = 10.87MHz$ .

On the basis of the connection in each HI is  $f_{p1} = f_{c1}$ , and  $f_{p2} = f_{c2}$ , therefore it is  $f_{p1} = f_{p2} = 10.87MHz$ .

As a consequence, the synchronization time delay depends mainly on  $O_C$  according to (6.2). The greatest value of  $O_C$  is equal to  $1/f_c = 92ns$ . This result confirms that the previous measured value of  $T$  is compatible with the hardware characteristics of the HIs, and corresponds to the greatest value achievable. As concerning the variation of the synchronization time delay, by assuming  $u_f$  in the range  $[1, 10]10^{-7}Hz$ , by (6.8) is  $u_{\Delta T} = 1/f_c\sqrt{6}$ . In the case, the differences between the mean values of the  $f_{c1}$  and  $f_{c2}$  and their standard deviation cannot be considered negligible, the delay is not constant according to the mathematical model (6.10).



**Fig. 6.7:** Snapshot of the DSO to evaluate the delay between the trigger signals generated by the HIs, in the case  $f_{c1} = f_{c2}$ .

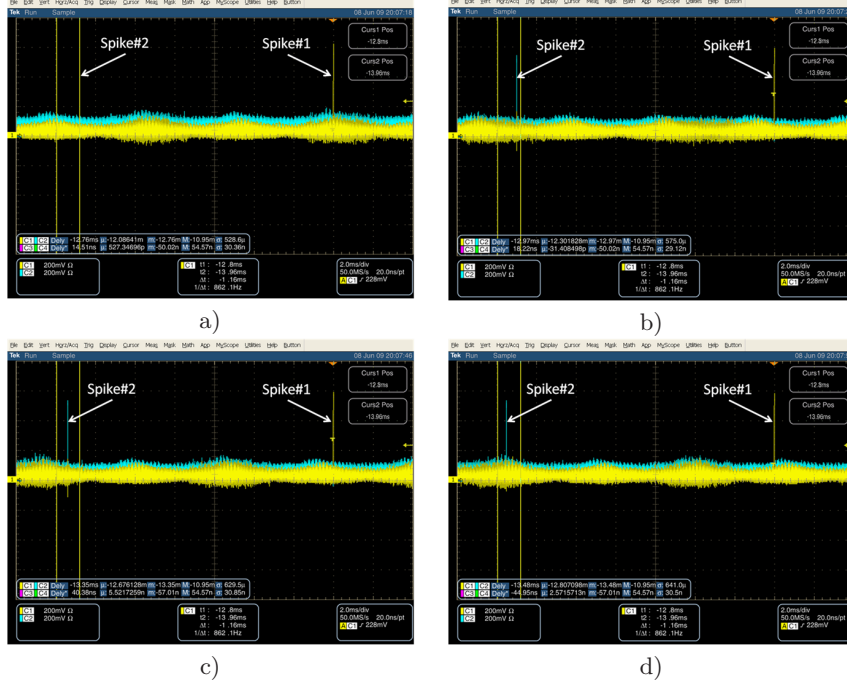
In particular, as shown in Fig.6.8a - d the difference between the mean value of  $f_{c1}$  and  $f_{c2}$  causes drift into the time delay synchronization.

In particular, the trigger signal generated by the PIC#1 is corresponding to the spike#1 in Fig.6.7. That one generated by the PIC#2 is corresponding to the spike#2) in Fig.6.7.

The Fig.6.8 a) - d) refers to increasing values of  $n_c$ : a)  $n_c = 574 * 10^7$ ; b)  $n_c = 582 * 10^7$ ; c)  $n_c = 600 * 10^7$ , and d)  $n_c = 607 * 10^7$ ). It can be noted as the spike#2) moves on the left respect to the spike#1. This effect agrees with the mathematical model (6.2).

In the case the differences between the mean values and the standard deviations of the  $f_{c1}$  and  $f_{c2}$  cannot be considered negligible, and/or the  $n_{c1}$  differs from  $n_{c2}$ , the functionality of the HI is changed.

Indeed, two HIs can be used to synchronize the MIs if the synchronization time delay and its variation have the minimum values, otherwise the HIs can be used to timing the MI with established time delay.



**Fig. 6.8:** Snapshot of the DSO to evaluate the synchronization time delay between the trigger signals generated by the HIs in the case  $f_{c1} \neq f_{c2}$  and a)  $n_c = 574 * 10^7$ , b)  $n_c = 582 * 10^7$ , c)  $n_c = 600 * 10^7$ , d) and  $n_c = 607 * 10^7$ .

## 6.7 Conclusions

A solution based on embedded HI is proposed in order to reduce the random causes of time delay in the detection of the trigger condition to start the command execution for the MI.

The hardware architecture and the logical operations performed by the proposed HI are discussed.

The model of the time delay occurring in the proposed HI is taken into account in order to detect the main causes that influence the synchronization time delay and its variation between two MIs.

Experimental tests have been performed in the case two HIs have the same clock frequency and the same trigger condition. The tests confirm the effectiveness of the proposed hardware solution, and the validity of the model of the synchronization time delay and its variation.

The operating conditions occurring when the oscillators used into the HIs have different frequencies and the trigger conditions are different are taken into account, also.

## Delivery the Common Sense of the Time by PDA to Stand Alone Measurement Instrument

*Abstract - It is proposed the use of the Personal Digital assistant (PDA) to delivery the common sense of time to the Measurement Instruments (MIs) not reachable, or not convenient to reach, by wired or wireless connections to the Distributed Measurement System (DMS). For this reason the standard synchronization protocol, or the propagation of a electromagnetic signal cannot be used. Therefore, once synchronized to the common reference node clock of the DMS, the PDA is used to physically bring the common sense of time to the inaccessible and stand alone MI.*

### 7.1 Introduction

The synchronization of the Measurement Instruments (MIs) connected to the Distributed Measurement System (DMS) needs that all the nodes involved must communicate to share the common synchronizing signal or message packets to adjust their clocks.

In some cases it is not possible to execute synchronized operations because:

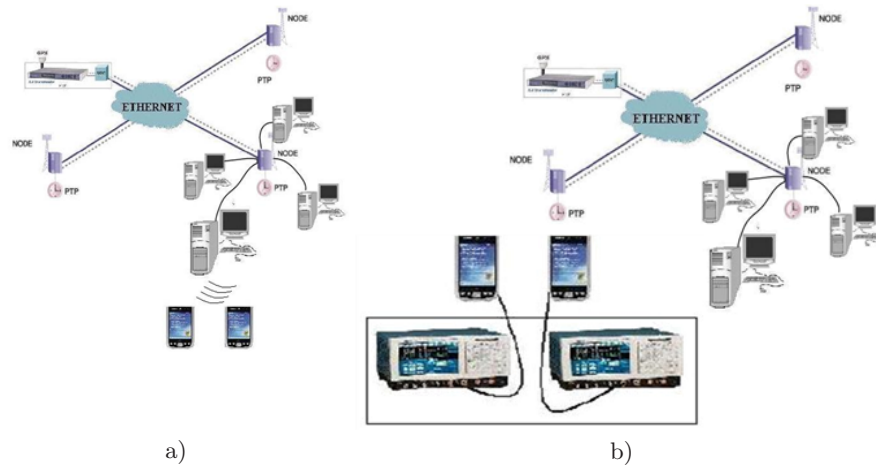
- a) the network's topology doesn't provides continuous, reliable and stable link, wired or wireless, to the whole network [141] -[130],
- b) the installation of new wired or wireless links or repeaters should be expensive or not physically or technically possible.

A solution is the use of embedded hardware in order to propagate the common sense of time by GPS signal [9] [71], [97], [2], [21], [138], [2], [106],[76]. However this solution can increase the cost of the system because requires the use of new, expensive and dedicated hardware [44] and cannot be applied in indoor measurement procedures as discussed in Chapter 2.

The increasing use of the Personal Digital Assistants (PDAs) makes possible to exploit these devices to drive and manage MIs or local DMS in synchronized manner to another MI or DMS, without wired or wireless connection to the previous ones [16],[49], [4], [3].

The use of the PDA to synchronize stand-alone MIs has the effect to extend the concept of Hardware Interface (HI) interfacing the MI to the node of the DMS. In this case the HI is constituted by the PDA in the place of the PC or the hardware embedded, considered in the research given into the previous chapters.

Therefore, the Master PC is previously synchronized into the DMS, through a boundary node (Fig.7.1). Successively, it extends the synchronization to the PDA with the suitable procedures.



**Fig. 7.1:** a) PDAs are synchronized to the synchronized node of the DMS, b) the PDAs bring the common sense of time to stand alone MIs.

In order to synchronize the PDA to the Master PC, the standard IEEE 1588 [68] can be taken into account. In this standard the protocol designed for the clock synchronization with microsecond or sub-microsecond accuracy is proposed. It is based on sharing the UDP packets marked with timestamp. Therefore, the accuracy of the time synchronization is depending on the accuracy of the timestamps.

In general, the PDA's technology provides the time resolution of the order of millisecond [142]. As a consequence the achievable time resolution for synchronization is multiple of the millisecond. This value is greater than the value of the sub-microsecond guaranteed by the standard IEEE 1588. This is a strong limitation to reach high accuracy in timestamp. Two solutions are proposed regarding the synchronization of the PDA to common reference clock, constituted by the Master PC, in order to bring "physically" the clock reference time to another MI or DMS inaccessible by the wired or wireless network.



Both the pointed out procedures are based on the count of the system's cycles in the place of the timestamp.

By assuming that the CPU frequency is of order of hundred MHz [30], the time resolution achievable with the use of system cycles is very close to multiple of the nanosecond.

The pointed out synchronization procedures take into account that the working frequency of the PDA can change respect to the nominal frequency of the CPU.

In particular the first solution is based on the self-evaluation of the working frequency of each PDA. On the basis of these values, each PDA evaluates the correction factor to change the number of system cycle to count in order to synchronize the operations to that of other PDAs.

The second solution is more complex and it is based on the optimized evolution of the previous one [52]. In particular, each PDA does not evaluate only one correction factor but the suitable look up table. This last is organized in order to best fit the behavior of the PDAs according to the number of cycle to count.

Experimental tests confirm the proposed strategies.

## 7.2 First synchronization procedure

The operations involving the Master PC, the PDA-Slave and the interactions among them on the basis of the proposed synchronization procedures are shown in Fig.7.2.

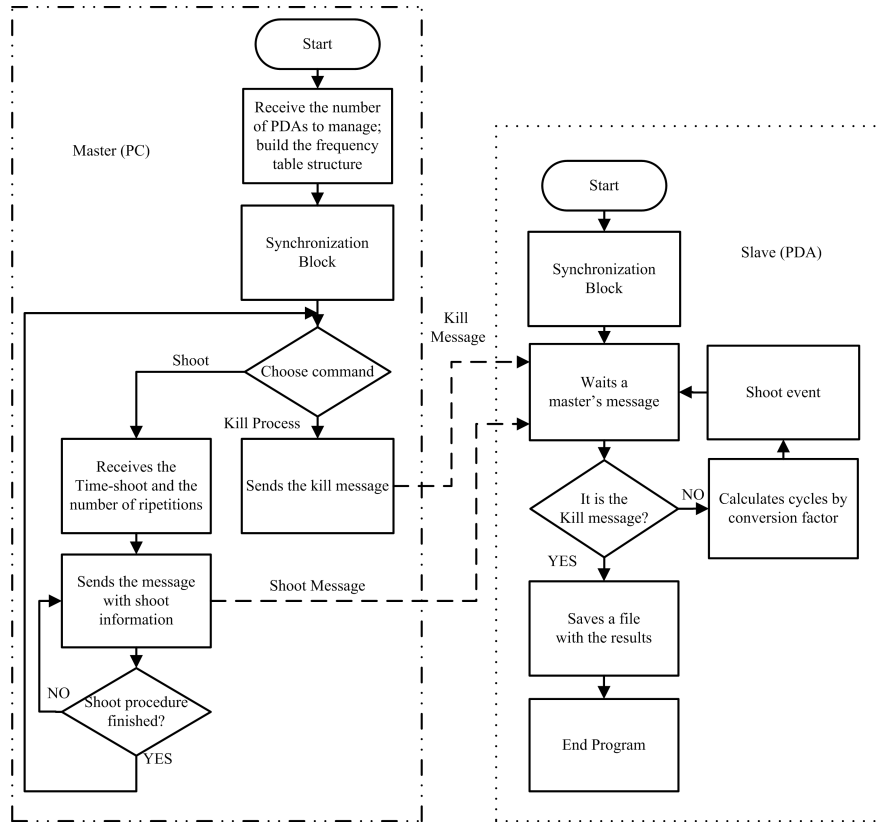
The first phase concerns with the synchronization between the Master and each one of all the available PDAs. This procedure is executed in the Synchronization block, according to the scheme shown in Fig.7.3

The PDA starts the synchronization procedure by evaluating the own actual clock frequency  $f_i$ . If the functions *QueryPerformaceFrequency* and *QueryPerformanceCounter* [89] are available, it is easy the evaluation of this frequency. Successively, the PDA sends the handshake message to the Master. The handshake message is necessary to permit the synchronization of only one PDA at time. The estimated value of  $f_i$  is sent to the Master and the PDA waits from the Master the reference frequency  $f_r$ .

The Master, after reception of  $f_i$  by the last PDA to be synchronized, sets the common reference frequency  $f_r$  by selecting the lower among the  $f_i$  received and sends it to all PDAs by using the broadcast WiFi message. The messages between Master and PDA are transmitted by Ad-Hoc WiFi network using the User Datagram Protocol (UDP).

Once received the reference frequency  $f_r$ , the PDA calculates the correction factor  $c_i$  on the basis of the following relation:

$$c_i = \frac{f_i}{f_r} \quad (7.1)$$



**Fig. 7.2:** Operations involving the Master PC, the PDA-Slave and the interactions among them, on the basis of the first proposed synchronization procedure.

The correction factor is used by the PDA to coordinate the execution time of the trigger message with all the others PDAs involved in the synchronized operations.

Finally, the PDA starts, at the higher priority [39], the thread to wait the shoot message or the kill message from the Master.

The idea to implement the synchronization on the basis of the system cycles introduces the problem of the different CPU frequencies among the Master and the synchronized PDAs. For this reason the common reference frequency  $f_r$  is selected and from (7.1) the value  $c_i$  is established. This value permits to evaluate the number of cycles  $n_i$  used by  $i$ -th PDA to execute the measurement in synchronized modality. Therefore, it is convenient that  $f_r$  is the lower among  $f_i$  to make  $c_i \geq 1$  for each PDA, and to reduce the uncertainty in the counting.

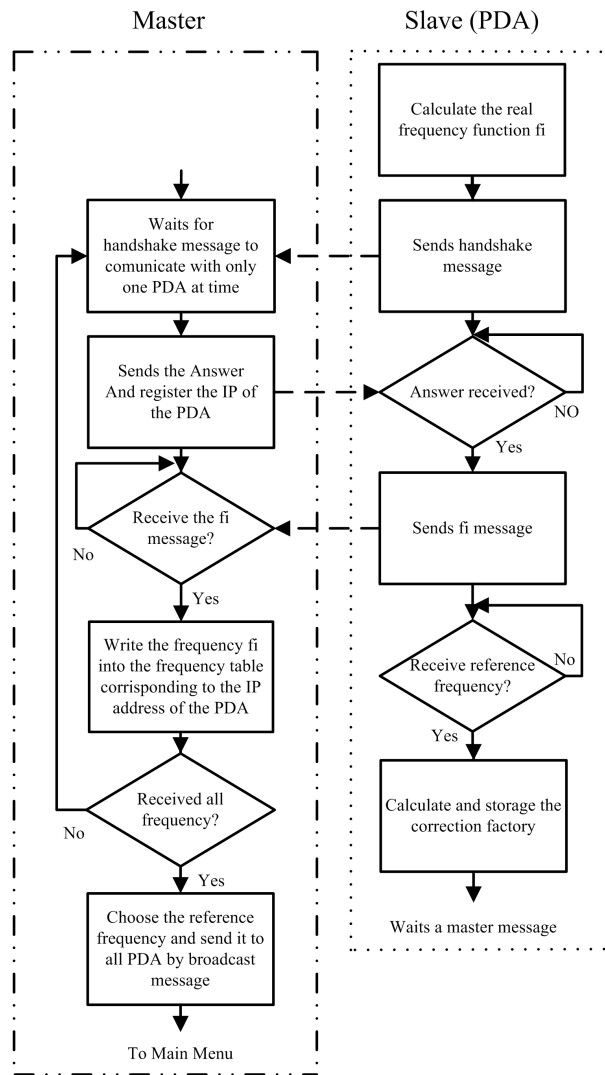


Fig. 7.3: Synchronization and unlock procedures.

The second phase concerns with the interaction among the Master and the PDA. In particular, the Master, after the synchronization procedure, can be configured in two different ways: (i) shoot mode, (ii) kill mode.

The shoot mode concerns with the setting of the shoot time in order to execute the measure. Through master interface the shoot time  $T_s$  is set by the user. The Master evaluates the number of system cycles  $n$  to transmit to all PDAs as:

$$n = T_s \times f_r \quad (7.2)$$

Once received the shoot message including the value of  $n$ , each PDA establishes the number of own CPU system cycle  $n_i$  as:

$$n_i = nc_i \quad (7.3)$$

Successively, the  $i$ -th PDA counts the number of system cycle  $n_i$ , and send the trigger message to the MI.

After forwarding the shoot message, the Master can return back to the main menu to wait a new commands from the user. The user can establish to execute a new measure by sending a new shoot message, or to finish all the operations by sending the kill message.

### 7.3 Outline of the optimized synchronization procedure

The main improvements achievable by the optimization of the synchronization procedure concern with:

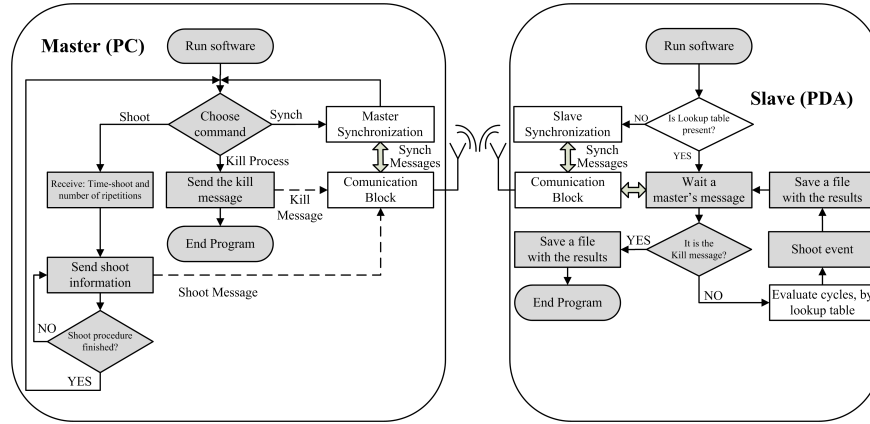
- *the reference clock.* It is not yet the slower clock of PDAs involved into the synchronization, but the node clock of the Master PC. In this way, a new PDA can be added into the group of the synchronized PDAs without repeat the synchronization for all the PDAs;
- *the reduction of the time required to synchronize all the PDAs.* The contemporaneous synchronization of all the PDAs involved, owing to the previous selection of the reference clock, reduce the time for the synchronization phase;
- *the introduction of the look-up table and the regression curve* to accurate evaluate the number of system cycles after whom send the command to the MI. In this way for each PDA the drift of the electronic components can be taken into account, and the accurate correction factor can be evaluated depending on the number of cycles to be counted.

As a result, the improved procedure merges the advantages of the synchronization obtained by applying the standard IEEE 1588 to the node clocks and the possibility to extend the common sense of time also in geographical area not reachable by the wired or wireless network.

#### 7.3.1 Synchronization phase

Fig.7.4 shows the block scheme of the interaction among the Master PC and the PDAs in order to perform the synchronization and to operate in synchronized modality. The grey colored blocks refers to the interaction among

the Master PC and the PDAs on the basis of the previous synchronizing procedure [8]. The Master PC can be configured in two different ways: syn-



**Fig. 7.4:** Block scheme of the interaction among the Master PC and the PDAs in order to perform the synchronization, and to operate in synchronized modality.

chronize mode, and shoot mode. In the synchronization mode the Master PC checks the presence of all the PDAs involved, and start the synchronization procedure. As in the previous procedure, the optimized one:

- is based on the count of the system cycle instead of the system time to perform the synchronized measure procedure in all the DMS nodes;
- the difference between the working frequency of Master PC and each PDA is taken into account.

The main difference between the optimized procedure and the previous one consists in the use of the *look up table*, instead of the *correction factor* to convert the number of system cycles sent by the Master PC  $n_c$  in the number of system cycles that the  $i$ -th PDA must count  $n_{ci}$ .

### 7.3.2 Operative phase

In the operative phase, the Master PC receives from the user the time in which the measure must be performed and how many time must be repeated. The Master PC converts the time in number of cycle  $n_c$  and sends this information to the PDAs. Each PDA evaluate the number of cycle to count  $n_{ci}$  on the basis of the own look up table previously evaluated in the synchronization phase. During the count of  $n_{ci}$  each PDAs can be moved in the different locations reaching the associated stand alone MI. Therefore the synchronized

measure procedure can be performed in all the DMS nodes at the established time and at the stand alone MI interfaced by PDA.

At the end of the procedure the user can chose to set a new synchronized procedure or end the working session. In this last case the Master PC sends the *kill message* to all the PDAs. In each PDA, the thread used to control the MI shut off, as received the *kill message*.

## 7.4 Working by the optimized synchronization procedure

In the first phase, the synchronization between Master PC and PDAs is performed. In particular, the  $i$ -th PDA creates the own lookup table useful to convert the number of cycles  $n_c$  received from the Master PC in  $n_{ci}$ , the effective number of cycles that it will count.

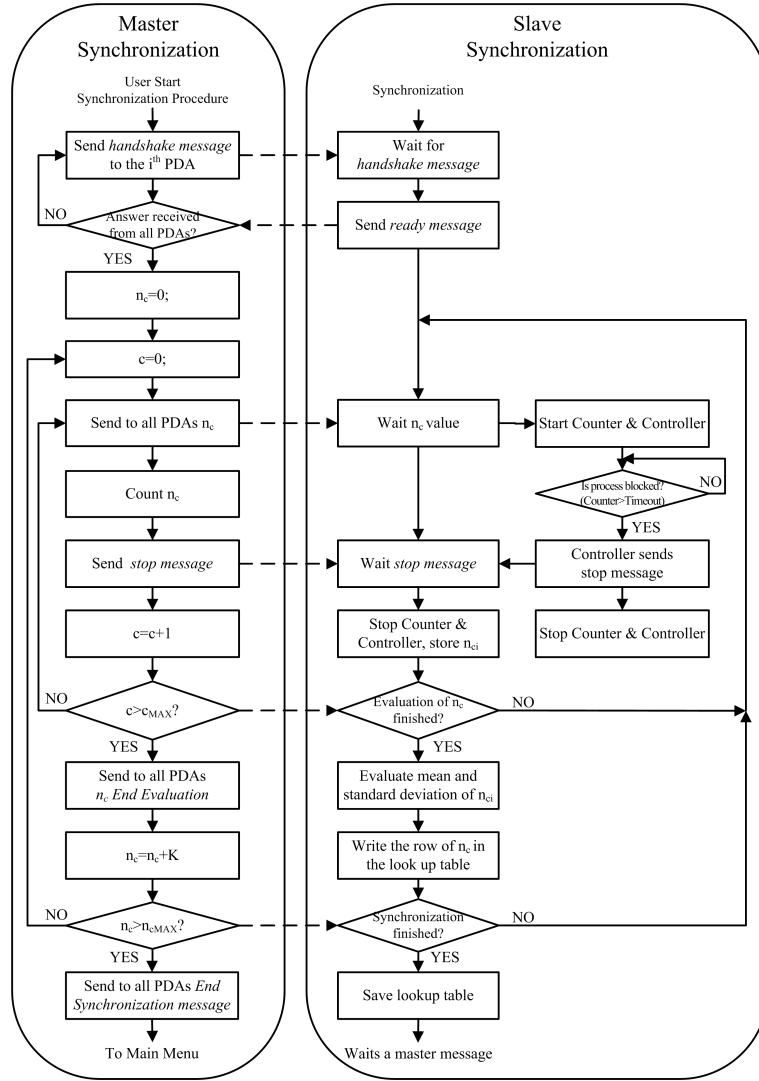
The block scheme of the synchronization procedure based on look up table is shown in Fig.7.5. In particular, Fig.7.5 explains the operations performed in the *Master Synchronization* block and *Slave Synchronization* block reported in Fig.7.1, and the interactions between them. After the handshake procedure, the Master PC sends to all PDAs, in broadcast modality, the number of cycles  $n_c$  to be counted. Successively, the Master and all the PDAs start the local counter. Once finished the count the Master sends the message to the PDAs that stop the counter and stores the cycle numbers of own counting. Therefore, the  $i$ -th PDA store the value  $n_{ci}$ . In this way the  $i$ -th PDA has the indirect information about the difference between the Master frequency and its one. This procedure is repeated with the same value of  $n_c$  several times to evaluate the mean value and the standard deviation of  $n_{ci}$ . The first row of look up table is constituted by: (i) the value of  $n_c$ , (ii) the mean value  $\mu_{ci}$  and (iii) the standard deviation  $\sigma_{ci}$  corresponding to  $n_{ci}$ . Other rows are constituted in similar manner by using different values of  $n_c$ .

During the synchronization procedure, the messages between the Master PC and the PDA are transmitted by using ad-Hoc WiFi network and the UDP. This is a protocol not connection oriented and it does not guarantee the package arrival, or ordered arrival, or the time in which they arrive. The absence of these controls helps speed transmission [24]. Based on this, sometimes, packets are lost, so it is necessary to introduce the mechanism of unlock. For this reason, when the PDA starts the synchronization procedure, more threads start:

- thread to count,
- thread to receive UDP packets (using synchronous socket [24]),
- thread to unlock slave, if blocked.

All these threads run with high priority.

Once completed the synchronization, the lookup table is stored in the root directory as *lt.txt* and *lt.dat* (*lt.dat* is necessary to load the same lookup tables



**Fig. 7.5:** Synchronization procedure based on look up table and unlock procedures.

in the future). Through the data stored into the lookup table the interpolation curve of the midpoints is calculated to increase the synchronization resolution.

Indeed, if the Master sends to the PDAs the number of cycles  $n_c$  not used in the synchronization process, each PDA is able to correct it on the basis of the Lagrange Interpolating Polynomial [121].

After the synchronization, the Master backs to the main menu to receive further commands, while the PDAs are ready to receive the shoot-message or the kill-message.

#### 7.4.1 Shoot mode

In the shoot mode, through the Master PC interface the shoot time is set and transmitted to the previous synchronized PDAs. Each PDAs, on the basis of the own lookup table, converts the data and executes the control procedure in order to perform the measure in the established time.

In this step all previous threads are killed in the PDA algorithm and only one, in the highest priority, is started. Once the shoot message is received, the PDA sends the WiFi packet or output signal to the stand alone MI after the number of cycles contained in the shoot message. If in the Master is chosen "end program", the Master sends to the PDA the message that kills the thread.

### 7.5 Experimental tests

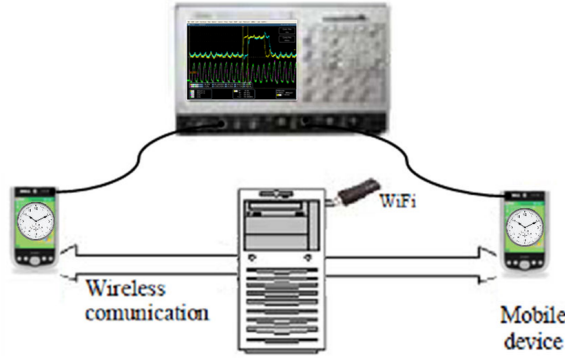
Experimental tests are executed by using two PDAs Dell X50v equipped with the same software. The aim of the first test is to highlight the lower value of accuracy achieved by the use of the time stamp into PDA and, successively, to investigate about the improvements achievable by the proposed synchronization procedure.

#### 7.5.1 Evaluation of the time stamp accuracy into PDA

The theoretical evaluation of the time stamp accuracy involves the analysis of the operations performed by the operating system of the PDA. As concerning with the interest of the research, this evaluation is performed by experimental approach. In particular, the DMS shown in Fig.7.6 is pointed out and used for the measurement of the time delay between the shoot time of the two PDAs.

Same control program based on the check of the PDA clock runs on each one. By operating on the Master PC, the user: (i) set the value of Start Time (ST) at which each PDA must generate the established signal on the earphone analog output, by sending the broad cast "Shoot message", and (ii) kills the test by sending the broad cast "Kill message". In the case the received message is "Shoot message" each PDA (i) saves the value of ST, (ii) checks in polling cycle the system time provided by the internal clock, and (iii) generates at the established time corresponding to TS the output signal. In the case the received message is "Kill message", each PDA aborts the execution of the program.





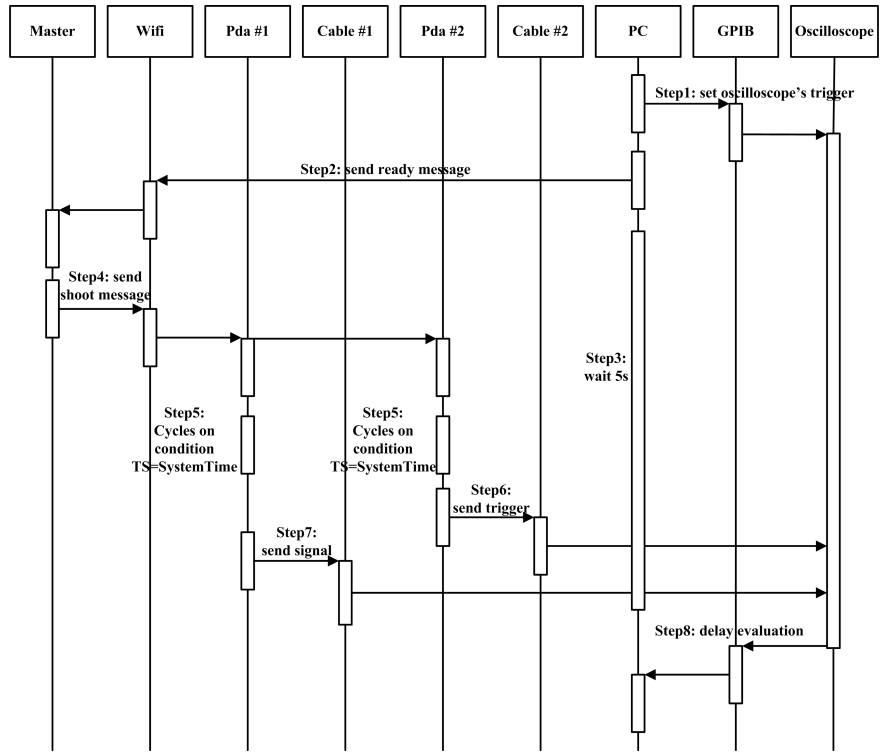
**Fig. 7.6:** Interaction among the two PDAs and the Master PC, and measurement of the time delay between the shoot time of the PDAs

The measure of the time delay between the two output signals is performed by using the Digital Storage Oscilloscope (DSO) Tektronix TDS7404 according to the planned command sequence of Fig.7.7.

Initially, the trigger of the DSO is set to acquire the incoming signals. Successively, the Master PC sends the broadcast packet to both the PDA Slave#1 and the PDA Slave#2. Both slaves count the common established number of cycles. When they finish to count, the PDA Slave#1 sends the signal to the trigger input of the DSO, and the PDA Slave#2 sends the signal to the channel #1. The delay is evaluated by the sample number occurring between the trigger and the cross of the established threshold by the signal of Ch1. Once set the trigger to 50% of the memory length, it needs to detect the first sample overcoming this threshold, only. In particular, denoted by  $i$  the half of the record length,  $j$  the index corresponding to the sample overcoming the threshold,  $dt$  the time between two samples, the time delay between the trigger and the signal is:

$$\Delta t = |(i - j) * dt| \quad (7.4)$$

In [8] it is shown that the shape of the output signal of the PDA influences the time delay value. In particular, the experimental test was performed to measure the synchronization time delay obtained when the electromagnetic signal is generated at the earphone output at the receiving of the broadcast WiFi message sent by the Master PC. Experimental tests highlight that by using the system "beep" the mean value of the time delay is  $\mu = 6.7 \cdot 10^{-4} s$  and the standard deviation is  $\sigma = 8.6 \cdot 10^{-4} s$ . By using the step signal constituted by 4.4 kSamples, the first 2.2 kSamples with value equal to zero and the second 2.2 kSamples with value equal to 250mV, is  $\mu = 2.0 \cdot 10^{-4} s$  and  $\sigma = 6.4 \cdot 10^{-4} s$ .



**Fig. 7.7:** Planned command sequence to evaluate the delay introduced by the two PDAs.

Therefore, the step signal is used in order to reduce the influence of the shape of the command signal on the time synchronization by PDAs.

The UDP packet loss is detected by taking into account that (i) if only one Slave is involved, it does not send the command to the DSO, and (ii) the Server receives the constant signals around the zero value. In the case both the Slaves are involved, the Server receives previous acquired signal.

If the two PDA clocks have the same frequency and are not synchronized, the time delay between the two clocks is observed in the experimental results as the mean value of the synchronization time delay different from zero. In this case, however, the interest of the test is devoted to evaluate the standard deviation. This last furnishes the accuracy of the timestamp obtainable by using the PDA clocks.

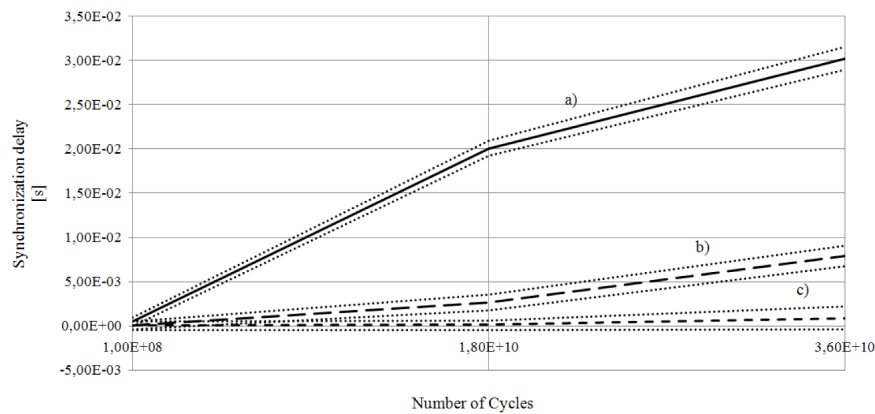
The performed experimental tests highlight that by analyzing the maximum range of  $TS = 5s$ , the maximum value of the synchronization time delay has mean value equal to  $4ms$ , and standard deviation equal to  $390ms$ .

These results highlight the low accuracy of the PDA clocks used into the test, and the latency on the access to the clock.

Therefore, by applying the synchronization technique based on *time stamped message passing*, the obtainable accuracy will be of the order of hundred ms, not suitable for the measurement purposes according to the standard IEEE 1588.

### 7.5.2 Accuracy achievable by the two proposed synchronization procedures

In order to assess the advantages of the proposed synchronization procedures, experimental tests were performed. The DMSs pointed out to execute the tests is similar to that described in 7.5.1. In particular, the PDA system cycles are used as time references instead of the internal clock according to the procedures described in the paragraphs 7.2 and in 7.3. In Fig.7.8 is shown



**Fig. 7.8:** Trend of the mean value and standard deviation of the synchronization time delay between the shoot time of the two PDAs versus the number of cycles (a) without the synchronization, (b) with the first synchronization procedure, and (c) with the optimized synchronization procedure.

the trend of the mean value and the standard deviation of the synchronization time delay between the shoot time of the two PDAs versus the number of cycles:

- a) without the synchronizing procedure;
- b) with the first synchronization procedure, described in 7.2;
- c) with the optimized synchronizing procedure, described in 7.3 and 7.4.

It can be noted that the synchronization time delay is practically constant once the PDAs are synchronized. In the contrary, without the synchronization the time delay increases.

By comparing the curve b) and c), it can be noted that the introduction of the look-up table and the regression line permit to reduce the mean value of the synchronization time delay.

## 7.6 Conclusions

In the chapter is proposed the use of the Personal Digital Assistant (PDA) to share the common sense of time to stand alone Measurement Instruments (MIs).

Once synchronized to the common reference node clock of the Distributed Measurement System (DMS), the PDA is used to physically bring the common sense of time to stand alone MI.

Two procedures to synchronize the PDAs to the Master node of the DMS are pointed out. Both the synchronization procedures are based on the system cycle in the place of the system frequency.

The improvements of the second optimized procedures respect to the first one [8] concern with the introduction of the look-up table and the regression line to establish the number of system cycles after whom to execute the control of the MI. In this way for each PDA the drift of the electronic components can be taken into account, and the accurate correction factor can be evaluated depending on the number of cycles to be counted.

The optimized procedure meets the advantages of the synchronization based on the standard IEEE 1588.

The results of experimental tests (i) validate the synchronization procedure pointed out, (ii) highlight the advantages, and (iii) justify the choice to perform the synchronization procedure on the basis of the system cycle in the place of the system frequency.

## Synchronizing Procedures to Delivery the Common Sense of the Time to Stand Alone Measurement Instrument by PDA and Embedded Hardware

*Abstract - In the previous chapter, the Personal Digital Assistant (PDA) is proposed to interface stand alone Measurement Instrument to the Distributed Measurement System. The achieved synchronization accuracy, is in the order of the  $\mu s$ . In this chapter is proposed an improved procedure based on the conjunct use of PDA and the proper designed Embedded Synchronizing Hardware to further reduce the time delay occurring in the measurement execution stage.*

### 8.1 Introduction

The optimized synchronization procedure, based on the Personal Digital Assistant (PDA) shown in the Chapter 7 permits to achieve synchronization accuracy in the order of  $\mu s$ .

In the case of stand-alone Measurement Instrument (MI), it is necessary not only propagate the common sense of the time to each MI, but also setting the parameters and collecting the data. These last operations are complex to implement into the embedded hardware and is more complex to change the measurement procedure.

Therefore, in the chapter is proposed the conjunct use of the PDA and the Embedded Synchronizing Hardware (ESH) to correlate in time the measurements of the stand alone MI to the Distributed Measurement System (DMS). The ESH used is based on the embedded hardware, presented in Chapter 6, that permits to achieve the sub- $\mu s$  accuracy.

By referring to the extended concept of Hardware Interface (HI) connecting the MI to the node of the DMS introduced in Chapter 7, the conjunct use of the PDA and ESH make it more extensive.

Moreover, by the mathematical model of the variation of the synchronization delay pointed out, more information are obtained to optimize the design of the ESH.

Experimental tests validate the presented HI and assess that this solution meets the advantages of the synchronization based on the standard IEEE 1588.

## 8.2 Description of the proposed extended HI

Fig.8.1 shows the practical use of the extended HI to perform synchronized measures also in the cases of stand-alone MIs. The extended HI is composed by the PDA and the ESH.

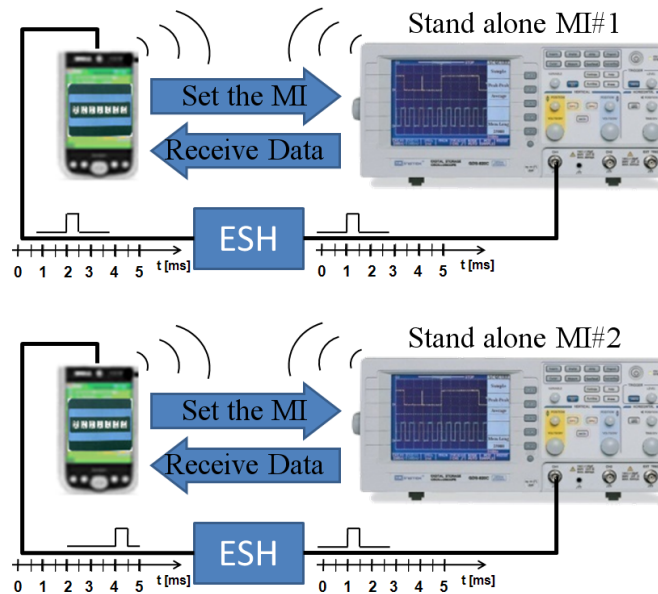


Fig. 8.1: Timing of the output signal of PDA#1, PDA#2, ESH#1, ESH#2.

The PDA executes the control of the MI and the storage of the data acquired by using the WiFi interface. In this way the software can be:

- implemented with high level languages as LabView, Java, Visual C++ etc.;
- easily changed and customized,
- equipped with Graphical User Interfaces (GUI).

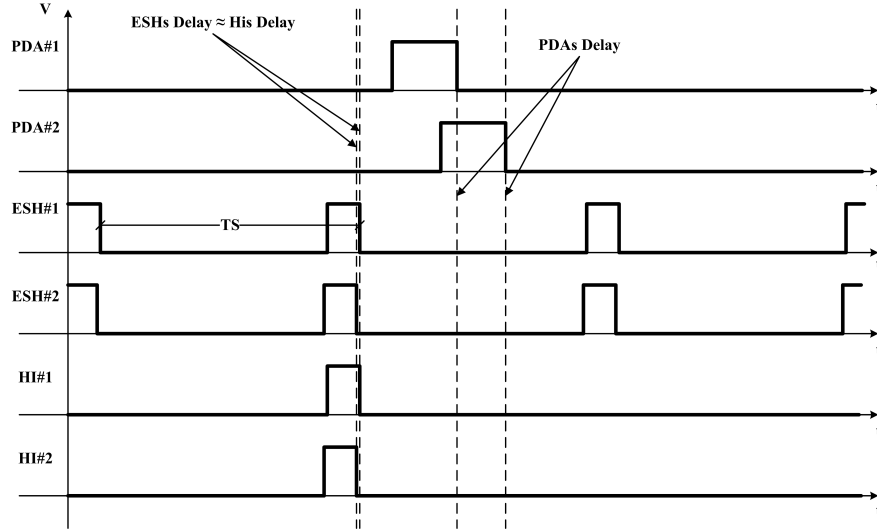
### 8.2.1 Theoretical criteria to design ESH

The ESH is used to synchronize the signal generated by the PDA with  $sub - \mu s$  accuracy.

It generates the periodic signal with period  $TS$ . This signal is synchronized with one generated by the other ESHs belonging to the other HIs.

In particular, the HI#1, constituted by the PDA#1 and the ESH#1, sends the signal to control the MI at the first Positive Slope of the Signal (PSS) of the ESH#1 after the PSS generated by the PDA#1.

In this way the synchronization delay between the PSSs generated by two HIs depends only on the delay between the PSSs generated by their ESHs, as shown in Fig.8.2



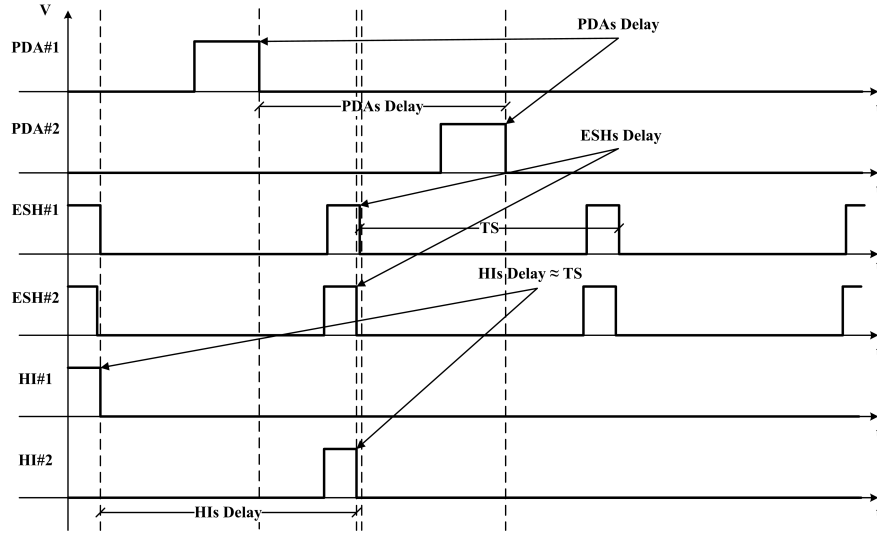
**Fig. 8.2:** Timing of the Output signal of PDA#1, PDA#2, ESH#1, ESH#2, HI#1, HI#2.

In particular, Fig.8.2 shows the time sequence of the signals generated by the PDA#1 and PDA#2, ESH#1, ESH#2, HI#1, and HI#2 involved in the synchronized measurement procedure of two stand alone MIs.

Fig.8.3 highlights that if the PSSs generated by the PDAs occur one before and another after the two PSSs generated by the ESHs the worst case occurs, and the delay between the PSSs generated by the HIs is equal almost to  $TS$ .

In order to avoid this last case, the  $TS$  value must be chosen by taking into account: (i) the variation of the delay between the PSSs generated by the PDA#1 and PDA#2 ( $u_{\Delta PDA}$ ), and (ii) the variation of the delay between the PSSs generated by the ESH#1, and ESH#2.

In particular, if the synchronization time delay between the PSSs generated by ESHs can be considered negligible respect to the one of the PDAs and to the value of  $TS$ , the probability of the worst case is:



**Fig. 8.3:** Timing of the output signal of PDA#1, PDA#2, ESH#1, ESH#2, HI#1, HI#2. If the PDAs signals rises one before and the other after the rising signal of the two corresponding ESHs the worst case occurs and the HIs signals delay is equal at most to TS.

$$p_{wc} = \frac{u_{\Delta PDA}}{TS} \quad (8.1)$$

Therefore, once established the accepted value of  $p_{wc}$ , evaluated the value of  $u_{\Delta PDA}$  it is possible to calculate the TS value by:

$$\Delta PDA = t_{PDA\#1} - t_{PDA\#2} \quad (8.2)$$

where  $t_{PDA\#1}$  is the time in which the PSS generated by the PDA#1 occurs,  $t_{PDA\#2}$  is the time in which the PSS generated by the PDA#2 occurs. These time are set by the master node according to the procedure described in 7.3.1.

In order to evaluate the  $\Delta PDA$  variation it can be noted that can be followed the procedure to evaluate the uncertainty in the case the quantity  $Y$  is determined from  $N$  independent measures  $X_1, X_2, \dots, X_N$  [57]:

$$Y = f(X_1, X_2, \dots, X_N) \quad (8.3)$$

Therefore, the law of propagation of uncertainty can be applied in order to establish the relationship among the  $\Delta PDA$  variation and the variation of the



above mentioned quantities [58]. In particular, it is:

$$f : \Delta_{PDA} = f(t_{PDA\#1}, t_{PDA\#2}) \quad (8.4)$$

and the variation of the synchronization time delay is:

$$u_{\Delta_{PDA}} = \sqrt{\left(\frac{\partial \Delta_{PDA}}{\partial t_{PDA\#1}}\right)_{t_{PDA\#1}, t_{PDA\#2}}^2 u_{t_{PDA\#1}}^2 + \left(\frac{\partial \Delta_{PDA}}{\partial t_{PDA\#2}}\right)_{t_{PDA\#1}, t_{PDA\#2}}^2 u_{t_{PDA\#2}}^2} \quad (8.5)$$

From (8.2) it is:

$$\left(\frac{\partial \Delta_{PDA}}{\partial t_{PDA\#1}}\right)_{t_{PDA\#1}, t_{PDA\#2}} = \left(\frac{\partial \Delta_{PDA}}{\partial t_{PDA\#2}}\right)_{t_{PDA\#1}, t_{PDA\#2}} = 1 \quad (8.6)$$

Therefore, the (8.5) can be rewritten as:

$$u_{\Delta_{PDA}} = \sqrt{u_{t_{PDA\#1}}^2 + u_{t_{PDA\#2}}^2} \quad (8.7)$$

The TS value valid for all the couple of PDA can be obtained by imposing that the  $u_{t_{PDA\#1}} = u_{t_{PDA\#2}} = \text{Max}(u_{t_{PDA\#i}})$ . Therefore the 8.7 can be rewritten as:

$$u_{\Delta_{PDA}} = \sqrt{2} \text{Max}(u_{t_{PDA\#i}}) \quad (8.8)$$

By referring to the optimized synchronization procedure proposed in paragraph 7.4,  $u_{t_{PDA\#i}}$  can be evaluated by taken into account: (i) the maximum standard deviation of the number of system cycles stored in the lookup table of the PDA#i ( $\sigma_{MAXn_{c_i}}$ ), and (ii) the working frequency ( $f_i$ ). If the functions *QueryPerformaceFrequency* and *QueryPerformanceCounter* [89] are available, it is easy the evaluation of this frequency as discussed in [8].

Therefore, the variation range of the time delay of the i-th PDA is:

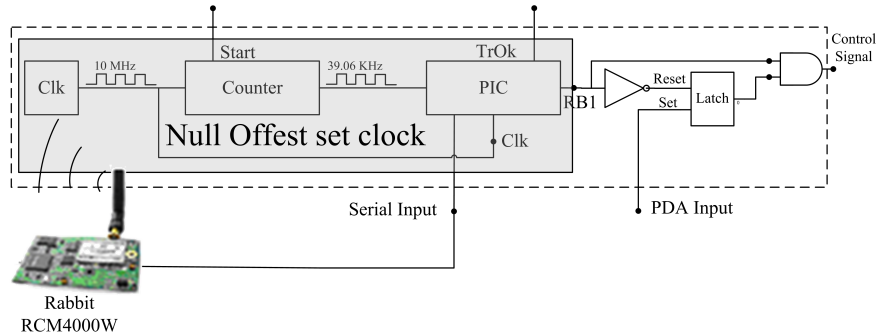
$$u_{t_{PDA\#i}} = \text{sigma}_{MAXn_{c_i}} f_i \quad (8.9)$$

Obviously, lower values of  $u_{\Delta_{PDA}}$  requires lower value of TS in order to achieve the same probability  $p_{wc}$ . Thus, it is justified the use of the procedure described in 7.4.

This consideration justify the study devoted to reduce the standard deviation of the synchronization delay between PDA signals. Indeed if the standard deviation of the delay between the PDA is in the order of 300 ms, and the accepted  $P_{wc}$  is 0.1%, the corresponding value of TS is about 424s.

### 8.2.2 Actual realization of the ESH

On the basis of the performed operation, the hardware architecture of the ESH is shown in Fig.8.4.



**Fig. 8.4:** Block scheme of the proposed ESH architecture.

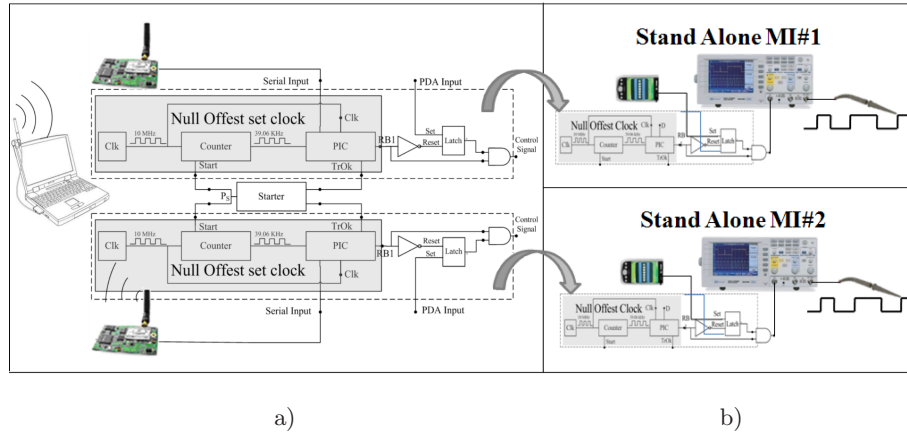
It is constituted by: (i) Null Offset Set Clock (NOC) [55], (ii) latch, and (iii) logical port.

The NOC is similar to the hardware interface presented in Chapter 6. It is used to generate the synchronized signal with one generate by the NOC of another HI with accuracy in the order of sub-microsecond. The parameters characterizing the output signal are established by the Master node and sent to the ESH by using the board Rabbit 4000W as WiFi-Serial protocol converter.

The other components, latch and logical port, are used to synchronize the signal generated by the NOC with the one generated by the PDA. Indeed, according to the block scheme of Fig.8.4, the PSS of the PDA set the latch on "true", this logical value is sent to the input of the "and" logical port. The output of the "and" is true only after the incoming of the PSS of the NOC to the other input of the "and". This last also rest the latch. Therefore the PSS of the output signal of the "and", sent to the external trigger of the MI, is consequence of the first PSS of the ESH that occurs after the PSS of the PDA.

### 8.3 Use of the extended HI

Fig.8.5.a) shows the block scheme of the proposed architecture in the configuration corresponding to the first phase in which the synchronization between the ESH is performed. The board Rabbit RCM4400W [109] is used to



**Fig. 8.5:** Block scheme of the proposed ESH architecture in the configuration corresponding to a) the First Phase in which the synchronization is performed and b) the second phase in which the HIs are used to interface the MI in order to perform the synchronized measure.

receive the information from the Master node of the DMS on the WiFi interface and retransmits it on the serial interface to the PIC of the NOC [55]. On the PIC the software runs to check the fire time and to send, by the pin RB1, the periodic signal to the *not* operator connected to the *reset port* of the latch.

In the second phase, shown in Fig.8.5b), the ESH is used to drive the MI in order to perform the synchronized measure with other DMS nodes.

In this phase the *set port* of the latch receives the PDA signal. The output of the latch is at high level and is in *and logic* with the PIC periodic signal. In this way the control signal is high only at the first rise of signal produced by the NOC that occurs after the rise of the signal generated by the PDA.

The external oscillator, temperature compensated, is used as reference clock for both the PIC and the counter, in order to temporize with high accuracy the commands execution.

Fig.8.6 shows the block scheme of operations executed by the extended HI in the two phases above described.

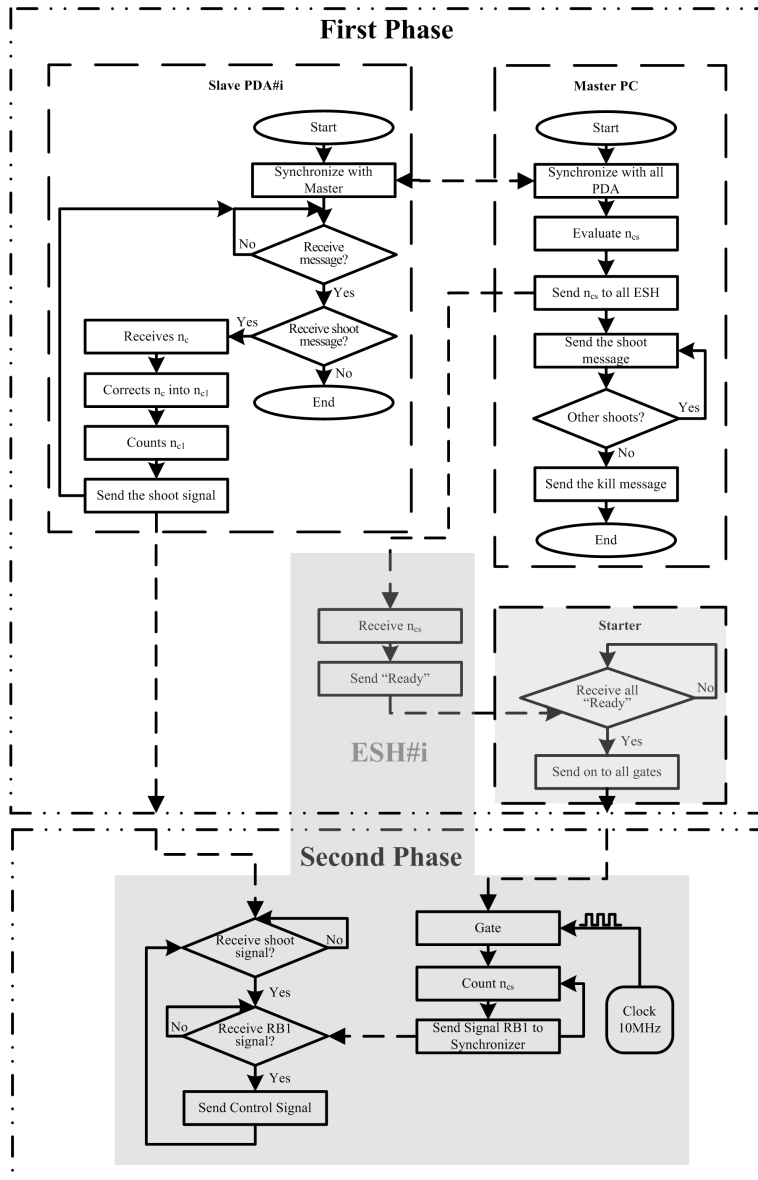


Fig. 8.6: Block scheme of the performed operation by the proposed ESH.

### 8.3.1 First phase: synchronization

The first phase concerns with the synchronization between the Master PC connected to the node of the DMS, the PDAs, and the ESHs.

The interaction between the Master PC and the PDAs is the same described in the paragraph 7.3.1. In particular in this first phase the Master PC synchronizes all the PDAs and transmits to all the PDAs the time TM in which the synchronized measure procedure must be performed.

The interaction between the Master PC and the ESHs consists in the transmission of the time period TS at which the NOC of each ESH must generate the signal on the pin RB1. This time is transferred in the form of *number of ticks* to count  $n_{cs}$ . It is evaluated by considering the frequency of the clock of the NOCs and the number of bits of the counter. The ticks, as shown in the block scheme of Fig.8.4, are generated by the counter and received by the PIC. Once each NOC is ready to *start the time counting*, it sends to the Starter block the *grant to start* by rising the voltage level on the pin TrOk. Once received all the grants, the Starter block sends to all NOCs the *Start to count* signal in parallel modality, by rising the voltage on pin  $P_S$ . Therefore, the delay time with whom the NOC counters are enabled can be considered negligible. In each NOC, once the counter has counted  $2^n$  ticks of the clock, produces the PIC interrupt signal. The operations performed by the PIC are: (i) increment the internal variable  $i$ , and (ii) check if  $i$  is equal to  $n_{cs}$ . If this condition occurs, the PIC rise the voltage level of the pin RB1. Owing to the proposed NOC, the execution time of the polling cycle  $T_{pc}$  to check the strike time TM, is negligible and, consequently, the accuracy of the NOC signal period is increased. Moreover due to the Starter Block, the offset between the PSS generated by each NOC is characterized by mean value equal to zero. Indeed, it starts the counters of each ESH in parallel modality and only after received the grants of all the ESH according to the procedure described in paragraph 6.3.

### 8.3.2 Second phase: measurement execution

The second phase concerns with the execution of the measure at the time TM.

The second phase does not require connection between the proposed architectures. Then each one is disconnected by the Starter block and is used to bring the common sense of time to the stand alone MI [24] as shown in Fig.8.5b). The pin *Control Signal* is connected to the MI, and the earphone output of the PDA is connected to the *PDA input* of the latch block.

Once the PDA rise the voltage level on the earphone output, the latch is enabled. The output of the latch at high level is in *and* with the signal incoming from NOC. Therefore when the ESH rises the voltage level on RB1 also the *Control Signal* rises the voltage level. The negative slope of the ESH signal is used to reset the latch in order to make it not sensitive to the periodic

PSS generated by the NOC. Therefore the PSS generated at the pin *Control Signal* occurs at the first PSS generated by the NOC, that occurs after the PSS generated by the PDA.

#### 8.4 Model of the synchronization time delay

According to the architecture of the ESH shown in Fig.8.4 and the functional modality, the synchronization delay  $\Delta T$  between two nodes of the DMS is:

$$\Delta T = 2^n n_{cs} \left( \frac{1}{f_{c1}} - \frac{1}{f_{c2}} \right) + n_p \left( \frac{1}{f_{p1}} - \frac{1}{f_{p2}} \right) + O_C + O_P; \quad (8.10)$$

where  $f_{c1}$  and  $f_{c2}$  are the frequency of the oscillators of the first and the second ESH, respectively,  $f_{p1}$  and  $f_{p2}$  the oscillator frequency of the PIC#1 and PIC#2,  $2^n n_{cs}$  the number of clock cycles counted in each ESH,  $n$  the number of bit of the counters;  $O_C$  the start offset time between the two counters, and  $O_P$  the start offset time between the two PICs;  $n_p$  the number of clock cycles needs to execute the control procedures in each PIC.

In the time interval corresponding to  $n_p$ , the following operations are performed:

1. the internal counter is incremented,
2. the occurrence of the trigger condition is checked,
3. the trigger procedure is executed.

The variation of the synchronization time delay evaluated from (8.10) is:

$$u_{\Delta T} = \sqrt{4^n n_{cs}^2 \left( \left( (-f_{c1}^{-2})|_{\bar{x}} u_{f_{c1}} \right)^2 \right) + \left( \left( (-f_{c2}^{-2})|_{\bar{x}} u_{f_{c2}} \right)^2 \right) + u_{O_C}^2 + n_{cs}^2 \left( \left( (-f_{p2}^{-2})|_{\bar{x}} u_{f_{p2}} \right)^2 \right) + \left( \left( (-f_{p1}^{-2})|_{\bar{x}} u_{f_{p1}} \right)^2 \right) + u_{O_P}^2} \quad (8.11)$$

where  $u_{f_{c1}}$ ,  $u_{f_{c2}}$ ,  $u_{O_C}$  and  $u_{O_P}$  are the uncertainties of  $f_{c1}$ ,  $f_{c2}$ ,  $O_C$  and  $O_P$ , respectively.

The (8.11) is valid both in absence of conditional jump and of correlation between parameters. Indeed, the conditional jump modifies the number of operations,  $n_p$ , to be executed on the basis of the particular event.

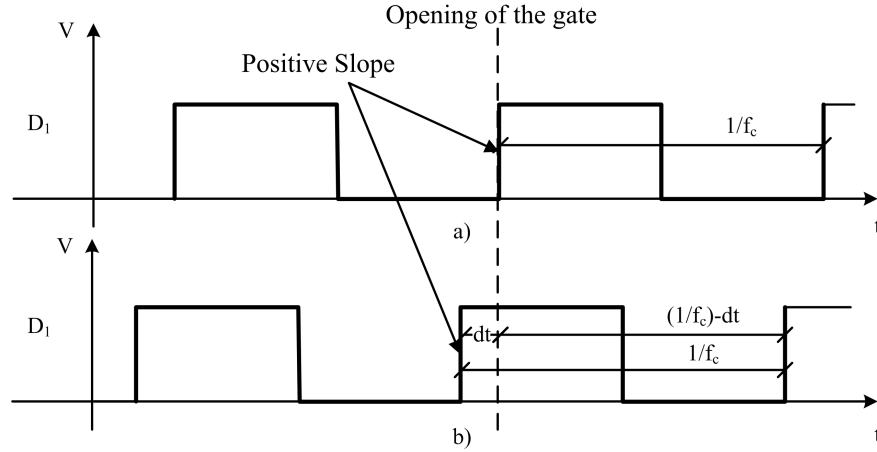
The offset  $O_C$  depends on the offset between the signals fed by the two clock blocks used as input to the two Counter blocks at the start of the count. If these two signals are not synchronized, and the Counter block is sensible only to the positive slopes of the input signal,  $O_c$  is the time delay between their positive slopes.

Therefore, considering negligible the delay between the time opening of the doors of the counters in HI#1 and HI#2,  $O_c = D_1 - D_2$ , where  $D_1$  is the

delay between the start of count and the incoming of the first positive slope of the signal incoming from clock on HI#2,  $D_2$  is the delay between the start of count and the incoming of the first positive slope of the signal incoming from clock on HI#2.

In the follow it is assumed that the value of  $D_1$  and  $D_2$  is distributed uniformly in the range  $[0 - f_c^{-1}]$ .

Indeed, as shown in Fig.8.7, taking into account only one signal, the delay  $D_1$  can be equal to zero, Fig.8.7a); or distributed between 0 and  $f_c^{-1}$ , as shown in Fig.8.7b). Fig.8.7 a) shows the case in which the positive slope of the signal



**Fig. 8.7:** Delay between the opening of the gate in the counter and the effective start to count. In a) the best case occurs, and the delay is zero. In b) the middle case occurs and the delay is equal to  $\frac{1}{f_c} - dt$ .

occurs just before the gate is opening. Fig.8.7 b) shows the case in which the positive slope of the signal occurs after  $dt$  the gate is open. Therefore, the counter will count the first positive slope after  $f_c^{-1} - dt$ . The worst case is for  $dt \rightarrow 0^+$ .

Therefore, according to the Guide to Uncertainty Measure [59] it is:

$$u_{O_C} = \sqrt{\left(\frac{\partial O_C}{\partial D_1}\right)_{D_1, D_2}^2 u_{D_1}^2 + \left(\frac{\partial O_C}{\partial D_2}\right)_{D_1, D_2}^2 u_{D_2}^2} \quad (8.12)$$

Because  $\left(\frac{\partial O_C}{\partial D_1}\right)_{D_1, D_2} = \left(\frac{\partial O_C}{\partial D_2}\right)_{D_1, D_2} = 1$  the (8.12) can be rewritten as:

$$u_{O_C} = \sqrt{u_{D_1}^2 + u_{D_2}^2} \quad (8.13)$$

Because  $u_{D_1}$  and  $u_{D_2}$  are evaluated on the basis of the uniform distribution, this case is similar to the case of uncertainty of type B. Therefore, in order to be used in the (8.13), they must be divided by  $\sqrt{3}$  [59]. Substituting the values of  $u_{D_1}$  and  $u_{D_2}$  in (8.13) and reducing the equation in its simplest form it is:

$$u_{O_C} = \sqrt{2 \left( \frac{1}{2\sqrt{3}f_c} \right)^2} = \frac{1}{\sqrt{6}f_c} \quad (8.14)$$

The start offset  $O_P$  can be neglected in the condition that the initialization procedure ends before the incoming of the first interrupt signal generated by the counter. This condition can be achieved by setting the values of  $n_p$ ,  $f_p$ ,  $n_{cs}$ ,  $f_c$ . Because it is necessary to wait the interrupt signal before to increment the counter internal to the PIC, all the operations performed before to execute the control procedures in the time interval corresponding to  $n_p$  are not influencing the variation of the synchronization time delay (8.11). Differently, this variation is depending on the operations performed in the time interval corresponding to  $n_p$ . Equation (8.11) suggests that to reduce the variation of the synchronization time delay, it is necessary to increase the accuracy of the clock frequency of the PIC and the counter.

## 8.5 Experimental tests

Experimental tests are executed in order to validate the HI and the model of the variation of the synchronization time delay pointed out in the paragraph 8.4.

Beginning, the two HIs are synchronized according to the block scheme of the Synchronization Phase shown in Fig.8.5.a). Successively, the two HIs are disconnected from the external Starter block, moved to the measurement place, and made available for the Operative Phase according to Fig.8.5.b).

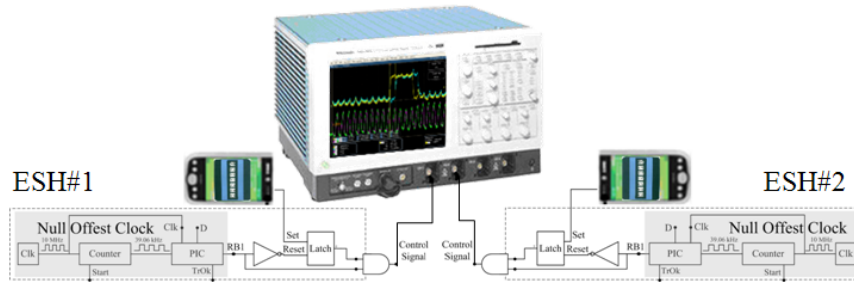
Because the intent is to detect and measure the time delay between the trigger signals furnished at the pin *Control Signal* of the HI#1 and HI#2, each one is connected to the input channel of the DSO, as shown in Fig.8.8.

The Master PC is connected by GPIB interface bus to the DSO, and sends the trigger condition to the two HIs by WiFi connection. The trigger signal fed from the HI#1 is sent to the channel#1 of the DSO, that of the HI#2 is sent to the channel#2 of the DSO.

The time delay between these two trigger signals furnishes the evaluation of the delay caused by the hardware and the software of the two HIs.

The delay is evaluated by means of the number of samples occurring between the cross of the established threshold by the two input signals. In partic-





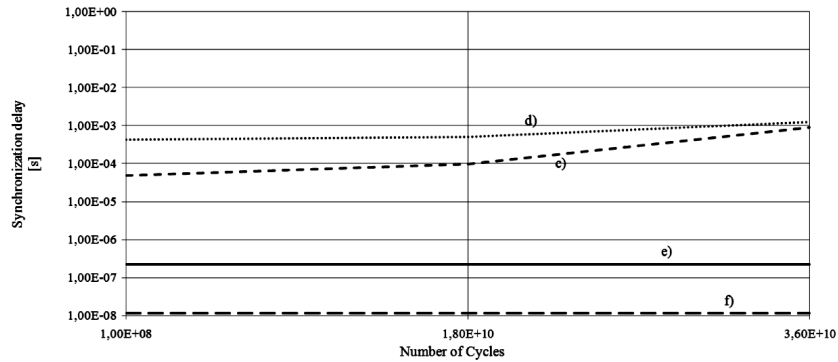
**Fig. 8.8:** Experimental set up to evaluate the delay between the trigger signals generated by the two HIIs.

ular, denoted by  $i$  and  $j$  the indexes corresponding to the samples overcoming the threshold,  $dt$  the time between two samples, the time delay is:

$$\Delta t = |(i - j) * dt| \tag{8.15}$$

Fig.8.9 shows the trend of the delay obtained by using the PDA and the synchronization technique based on the lookup table according to paragraph 7.4, and the ones obtained by using the PDA and the ESH.

In particular, the curve (c) refers to the trend of the mean value of the delay obtained with the first technique and the curve (d) is the trend of the standard deviation; the curves (e) and (f) are the trend of the mean value and standard deviation of the delay obtained by using the proposed ESHs.



**Fig. 8.9:** Trend of the mean value (c) and standard deviation (d) of the synchronization delay between the shoot time of the two PDAs applying the correction factor obtained by the lookup table versus the number of cycles. Trend of the mean value (e) and standard deviation (f) of the synchronization delay between the shoot time of the two PDAs and the proposed ESHs versus the number of cycles.

The experimental results confirm the mathematical model pointed out in the paragraph 8.4. Indeed, the synchronization accuracy in the order of sub microsecond is achieved, and the independence of the synchronization accuracy by the number of cycle to count is demonstrated.

## 8.6 Conclusions

In the chapter is presented a new extended HI to synchronize the stand alone MI to the node clock of DMS. It is constituted by the conjunct use of the PDA and ESH. The PDA permits the advantages of the high level programming languages in order to interface the MI and collect the data. The ESH guarantees the synchronization time accuracy owing to the deterministic behavior of the hardware architecture. Both the mathematical model of the time delay variation and the experimental tests confirm the suitability of the proposed solution to the requirements of the standard IEEE 1588.

---

## Conclusions and Direction for Future Research

Synchronized measurement is a new prospective service achievable by Distributed Measurement Systems (DMSs).

The research is devoted to analyze the aspects concerning with the synchronization of the co-operating Measurement Instruments (MIs) into the DMS.

Usually, the MI is interfaced to the node of the DMS by Hardware Interface (HI) and the synchronization procedure operates on the clock, internal or external, as discussed in the Chapter 2. Therefore, the HI works in synchronized modality with the accuracy level granted by the particular synchronization method used. But the usual hardware and the software architectures of the path involved in the communication HI-MI make random variables of the time delay in the command execution. The final result is that the synchronization time delay among the MIs can be influenced, and the values of mean and standard deviation increase.

In Chapter 3 it was proved that if the HI is constituted by PC, the Operating System (OS) is the main cause of the random variation of the time delay of the command execution, and the contribution of the MI's hardware is negligible. Indeed, the modern OS is multipurpose and multiprocessing, and the concurrency among processes makes random variables of the execution time of the command and then the time synchronization among MIs.

Different strategies were proposed to reduce the effects of random causes depending on the concurrency of the OS.

In order to investigate the effects of the non predictive software process on the communication delay the Linux OS is employed. Three different strategies to set up the Linux OS to reduce both the mean value and the standard deviation of the synchronization time delay were proposed in the Chapter 4.

The first strategy concern both the reduction of the number of the concurrent processes and the increase of the priority of the interesting process. Consequentially, the probability that another concurrent process can request the CPU is reduced. Therefore, the kernel makes available more computational

resources for the interesting procedure devoted to the command execution to send to MI.

The second strategy concerns the execution in kernel mode of the interesting procedure devoted to the command execution, by modifying the driver of the WiFi-USB adapter. Operating in this manner, the naturally priority of the kernel is overworked.

The third strategy concerns the use of the Linux Real Time OS. Because the Real Time OS grants the upper bound of the execution time of the interesting procedure, the advantage is the reduction of the mean value and the standard deviation of the time delay.

The choice among the above-mentioned strategies depends on the final accuracy required to synchronize the MIs.

Experimental tests have highlighted that each solution is characterized by a different lower bound of the time delay and different implementing and configuring complexity.

In particular, the solution based on proper setting up of Linux OS permits the synchronization accuracy of the order of  $\mu s$ . That solution based on the employment of Linux OS RT permits the synchronization accuracy of the order of  $0.1\mu s$  and that based on the software driver modification of the WiFi-USB receiver permits the synchronization accuracy of the order of  $\mu s$ .

Other experimental tests have assessed the independence of the synchronization accuracy from the architecture of the MI.

A different solution was proposed in Chapter 5. The new solution is based on the use of the Programmable Logic Device (PLD) in the place of PC as HI. The advantage to this solution is that the concurrency causes among the software processes are completely avoid, and, consequently, the execution time of the interesting procedure devoted to the command execution for the MI should be deterministic.

Nevertheless, operating conditions can occur making the detection of the trigger condition to start the command execution undeterminable. Usually this trigger is: (i) the incoming signal, wired or wireless, provided by the master node, or (ii) the previous established fire time.

The PLD detects the trigger only after checking the condition. If the trigger occurs after the check, it will be detected at the next check. In this case, the final result is that the start time of the command execution is random and depends on the time delay between the trigger and the trigger check. This delay can be at the most equal to the execution time of the polling cycle.

On the basis of this analysis the model of the variation affecting the synchronization time delay was pointed out. Owing to this model the effects of each cause affecting the trigger check was evaluated. Consequently, adequate strategy to reduce the random variation of the detection of the trigger condition was pointed out.

The proposed strategy concerns the reduction of: the time interval of the polling cycle, the uncertainty of the incoming trigger, the Start Offset and the ratio between the Start Offset and the execution time of polling cycle.

The experimental tests validated the proposed strategy.

In order to overcome the inconvenience arising from the random time delay of the start time of the MI, caused by the polling cycle, in Chapter 6 a different HI was taken into account to interface the MI to the node of the DMS. This new HI is designed and realized to reduce the random variation of the time delay and to avoid the polling cycle to check the trigger condition. The hardware architecture and the logical operations performed by the proposed HI were discussed in Chapter 6.

The model of the time delay occurring in the proposed HI is taken into account in order to detect the main causes that influence the synchronization time delay between two MIs. Experimental tests were performed where two HIs have the same clock frequency and the same trigger condition. The tests confirm the effectiveness of the proposed hardware solution, and the validity of the model of the synchronization time delay.

The scenarios of stand alone MIs cooperating to perform synchronized measures are also taken into account. In these cases it is not possible to use the synchronization techniques described in literature to synchronize the DMS nodes.

In Chapter 7 it is proposed that the Personal Digital assistant (PDA) is used to share the common sense of time to stand alone MIs.

The use of the PDA to synchronize stand-alone MIs has the effect to extend the concept of Hardware Interface (HI) connected to the MI. In this case the HI is constituted by the PDA in the place of the PC as considered in the previous chapter.

Once synchronized to the common reference node clock of the DMS, the PDA was used to physically bring the common sense of time to stand alone MI.

The suitable synchronization method pointed out was based on the system frequency in the place of the system time.

In order to take into account the hardware characteristic concerning the execution time of the counting operation involved in the synchronization procedure, the system's cycle of the PDA is considered.

The two procedures to synchronize the PDA to the DMS, and that procedure performed to the PDA to allow operation in synchronized manner versus the MI were described. The first is easy to implement, the second is the optimized version of the previous one.

The improvements concern the introduction of the look-up table and the regression line to establish the number of system cycles after which execute the control of the MI. In this way, for each PDA the drift of the electronic components can be taken into account, and the accurate correction factor can be evaluated depending on the number of cycles to be counted. The improved procedure meets the advantages of the synchronization based on the standard IEEE 1588.

The results of experimental tests (i) validate the synchronization method pointed out, (ii) highlight the advantages, and (iii) justify the choice to per-

form the synchronization procedure on the basis of the system frequency in the place of the system time.

In order to further reduce the delay introduced by the PDA, in Chapter 8 is proposed the conjunct use PDA and the Embedded Synchronizing Hardware.

By referring to the extended concept of HI connected to the node of the DMS and interfacing the MI, introduced in Chapter 7, the conjunct use of the PDA and Embedded Synchronizing Hardware make it more extensive.

The performed experimental tests grant that the achieved synchronization accuracy is in the order of 100 ns.

In conclusion, the research furnishes theoretical and practical indications to choose the suitable solution in order to satisfy the requirements of time synchronization accuracy in an effective and efficient way.

Many are the applications requiring synchronized measurement procedures, and many are the operational conditions in which they are performed. Therefore the ongoing activity regards with the characterization of the delay introduced by the interferences in the wireless communication between the nodes with the aim to setting up techniques and procedures to grant stable and reliable time synchronization.

Indeed, phenomenon such as packet loss occurring in the wireless network are not considered in the wired one. The packet loss is due principally to the high level of interference produced by concurrent networks, and noise. It is the cause of the synchronization performance reduction. This problem is not considered into the synchronization standard definition. Therefore, the complete research points out robust procedures to reduce the effect of the interferences on the synchronization service.

---

## References

1. J. Achkar, P. Tuckey, P. Ullrich, D.Valat, A. Batchelor, G. Burden, A. Bauch, D. Piester, F. Cordara, P. Tavella, J. Davis, J. Delporte, R. Jones, T. Levin, G. Staton, J. Nawrocki, J. M. Pieplu, "Fidelity-Progress Report on Delivering the Prototype Galileo Time Service Provider", Proceedings of the IEEE International Frequency Control Symposium, joint with the 21st European Frequency and Time Forum, May 29-June 1, 2007, pp.446-451.
2. M. Addouche, F. Meyer, F. Vernotte, "Uncertainty Estimation on Multi-Channel GPS Time Transfer", Proceedings of the IEEE International Frequency Control Symposium and Exposition, August 23-27, 2004, pp.351-355.
3. A. Aiello, D. L. Carni, D. Grimaldi, G. Guglielmelli, F. Lamonaca, "Distributed Measurement System Based on Wireless Mobile Device and Application Repository Server", Proceedings of the IEEE International Conference on Software in Telecommunications and Computer Networks, September 29-October 1, 2006, pp.162-166.
4. A. Aiello, D. L. Carni, D. Grimaldi, G. Guglielmelli, F. Lamonaca, "Wireless Distributed Measurement System Based on PDA and Dynamical Application Repository Server", Proceedings of the IEEE Instrumentation and Measurement Technology Conference, May 1-3, 2007, pp.1-6.
5. G. Algie, "Telecom Update to IEEE 1588 Workshop", in NIST publication NISTIR 7192, Proceedings of the Conference on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Gaithersburg, Maryland, September 27-29, 2004.
6. D. Allan and M. M. Weiss, "Accurate Time and Frequency Transfer During Common-View of a GPS Satellite", Proceedings of the Frequency Control Symposium, 1980, pp.334-336.
7. J. Balcells, V. Parisi, D. Gonzalez, "New Trends in Power Quality Measuring Instruments", Proceedings of the IEEE 28th Annual Conference of the Industrial Electronics Society, vol. 4, November 5-8, 2002, pp.3344-3349.
8. A. Barresi, D. Grimaldi, F. Lamonaca, "Delivery the Common Sense of the Time to Stand Alone Measurement Instrument by PDA", Proceedings of the IEEE I2MTC 2009, Singapore, May 5-7, 2009.
9. A. Bauch, D. Piester, A. Moudrak, G. Petit, "Time Comparisons Between USNO and PTB: A Model For The Determination of the Time Offset Between Gps

- Time and The Future Galileo System Time”, Proceedings of the IEEE International Frequency Control Symposium and Exposition, August 23-27, 2004, pp.334-340.
10. K. Behrendt, K. Fodero, ”The Perfect Time: An Examination of Time Synchronization Techniques”, Proceedings of the Distributech, Tampa, Florida, February 7-9 2006.
  11. J.C. Bellamy, ”Digital Network Synchronization”, IEEE Communications Magazine, vol. 33, N. 4, April 1995, pp.70-83.
  12. M. Bertocco, ”Architectures for Remote Measurement”, in Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements, Edit. F. Davoli, S. Palazzo, S. Zappatore, Springer, 2006, pp.349-362.
  13. A. Bottoni, ”Come Creare un Virus per Linux in 5 Facili Mosse (forse)”, March 24 2009, <http://alessandrobottoni.wordpress.com/2009/03/24/come-creare-un-virus-per-linux-in-5-facili-mosse-forse/>.
  14. G. Brundu, ”Torna Il Pinguino Del Web”, on Il Sardegna, 2009, p.38 <http://www.unica.it/pub/7/show.jsp?id=9608&iso=655&is=7>.
  15. D. L.Carni’, D. Grimaldi, G. Guglielmelli, F. Lamonaca, ”Synchronization of Measurement Instruments Co-operating into the W-Dms”, Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Warsaw, Poland, May 1-3, 2007.
  16. D. L. Carni, Grimaldi, D.; F. Lamonaca, A. Nastro, ”Customized Configuration and Management of Measurement Procedures on PDA”, Proceedings of the IEEE Instrumentation and Measurement Technology Conference Proceedings, May 12-15, 2008, pp.1037-1041.
  17. K. M. Carroll, T. Celano, ”Timing via the New LORAN-C System”, Proceedings of the IEEE International Frequency Control Symposium and PDA Exhibition, 2003, pp.245-249.
  18. S. M. Chamberlain, ”Combined GPS/GLONASS Navigation”, Proceedings of the National Telesystems Conference, Vol.1., NTC ’91, March 26-27, 1991, pp.205-210.
  19. Lin Chen, J. Leneutre, ”A Secure and Scalable Time Synchronization Protocol in IEEE 802.11 Ad Hoc Networks”, Proceedings of the International Conference on Parallel Processing, August 14-18 2006, p.8.
  20. M. Christer, S. Passe, S. Stein, ”Design of an IEEE-1588 Interface for Subnanosecond Performance”, Proceedings of the Proceedings of Conference on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Gaithersburg, Maryland, October 2-4, 2006.
  21. J. D. Clarke, J. A. Davis, A. J. Lowe, ”Characterisation of NPL’s Geodetic GPS Time Transfer Receivers”, Proceedings of the Frequency and Time Forum, joint with the Meeting of the European IEEE International Frequency Control Symposium, Volume 1, April 13-16, 1999, pp.225-229.
  22. W. J. Clinton, ”Statement by the President, Regarding the United States’ Decision to Stop Degrading Global Positioning System Accuracy”, White House, Office of the Press Secretary, Washington, D.C., May 1, 2000.
  23. J. Corbet, A. Rubini, G. Kroah-Hartman, ”Linux Device Driver third edition”, Stanford California USA, O’Reilly, 2005.
  24. Couch L, ”Digital and Analog Communication Systems 5th ed.”, Prentice Hall, 1997.



25. C. Curry, "Global Telecom Network Timing Standards-Decision Day Draws Close!", WSTS Review 2006, [http://www.chronos.co.uk/pdfs/tel/WSTS\\_Review\\_2006.pdf](http://www.chronos.co.uk/pdfs/tel/WSTS_Review_2006.pdf).
26. J. Dai, "Time Correlation Using Network Based Data Acquisition On-Board a Military Test Vehicle", in NIST publication NISTIR 7070, Proceedings of the Workshop on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Gaithersburg, Maryland, 24 September, 2003.
27. P. Daponte, D. Grimaldi, L. Nigro, F. Pupo, "Distributed Measurement Systems: an Object-Oriented Architecture And A Case Study", Computer Standards and Interfaces, vol.18, No.5, June, 1997, pp.383-395.
28. P. Daponte, D. Grimaldi, M. Marinov, "Real-Time Measurement And Control of an Industrial System Over the Internet: Implementation of a Prototype for Educational Purposes", IEEE Transaction on Instrumentation and Measurement, vol.51, N.5, October, 2002, pp.962-969.
29. D. Deeths and G. Brunette "Using NTP to Control and Synchronize System Clocks-Part I: Introduction to NTP", Sun BluePrints<sup>TM</sup> OnLine-July, 2001 available online at <http://www.sun.com/blueprints/0701/NTP.pdf>.
30. Dell<sup>TM</sup> Axim<sup>TM</sup> X50, Owner's Manual 2004.
31. B. Dickerson "Substation Time Synchronization", Protection, Automation and Control World-Summer, 2007, pp.39-45.
32. Dynamic C TCP/IP User's Manual Volume 1 USA, 2007, p.47.
33. J. C. Eidson, D. Pleasant, "Timescales and IEEE 1588-Part 2," LXI Connexion Magazine, October, 2006.
34. J. C. Eidson, M. C. Fischer, J. White, "IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", Proceedings of the 34th Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting, Reston, Virginia, USA (U.S. Naval Observatory, Washington, D.C.), 3-5 December, 2002, pp.243-254.
35. J. C. Eidson, "IEEE 1588: an Update on the Standard and Its Application", Proceedings of the 38th Annual Precise Time and Time Interval (PTI) Meeting, Reston Virginia, December 5-7, 2006, pp.193-211.
36. Y. K. Feng; J. X. Li, "Timing for the Loran-C Signal by Beidou Satellite and Error Correction for the Transmission Time", Proceedings of the IEEE International Frequency Control Symposium, May 19-21, 2008, pp.476-478.
37. G. Fortino, D. Grimaldi, L. Nigro, "Multicast Control of Mobile Measurement Systems", IEEE Transaction on Instrumentation and Measurement, vol. 47, No. 5, October, 1998, pp.1149-1154.
38. G. Fortino, D. Grimaldi, L. Nigro, "An Agent Based Measurement Laboratory Over Internet", Proceedings of the IEEE Autotestcon'99, S. Antonio, Texas, USA, August 30-September 2, 1999, pp.61-66.
39. N. Frampton, R. Lee, "Implementing Fault Tolerant Systems with WindowsCE", <http://www.windowsfordevices.com/articles/AT3859908246.html>.
40. Geirinhas Ramos H, "IEEE Standard 1451 and a Proposed Time Synchronization Approach", IEEE Instr. & Measur. Mag., Vol. 11, Is. 2, April, 2008, pp.29-37.
41. G. Gerten, "Protecting the Global Positioning System", IEEE Aerospace and Electronic Systems Magazine, Volume 20, Issue 11, November, 2005, pp.3-8.

42. M. Gilany, E.S.T. El Din, M.M. Abdel Aziz, D.K. Ibrahim, "An Accurate Scheme for Fault Location in Combined Overhead Line with Underground Power Cable", Proceedings of the IEEE Power Engineering Soc. General Meeting, vol. 3, June 12-16, 2005, pp.2521-2527.
43. G. Giorgi, C. Narduzzi, M. Stellini, "A Test Bed For Synchronization In Heterogeneous Network Environments", Proceedings of the IEEE IMTC/08, Victoria, Vancouver Island (Canada), May 12-15, 2008, pp.612-617.
44. GPS, Essentials of Satellite Navigation, Compendium, 2009, Document number GPS-X-02007-D
45. S. Granneman, "Linux vs. Windows, Viruses" consultable on [http://www.theregister.co.uk/2003/10/06/linux\\_vs\\_windows\\_viruses/](http://www.theregister.co.uk/2003/10/06/linux_vs_windows_viruses/) at August 08, 2008.
46. D. Grimaldi, L. Nigro, F. Pupo, "Java Based Distributed Measurement Systems" IEEE Transaction on Instrumentation and Measurement, vol. 47, No. 1, February, 1998, pp.100-103.
47. D. Grimaldi, M. Marinov, "Distributed Measurement Systems", Measurement, 30, 2001, pp.279-287.
48. D. Grimaldi, "Workload Measurements and Synchronization into a Distributed Measurement Laboratory", Measurement, vol.39, N.7, 2006, pp.655-663.
49. D. Grimaldi, F. Lamonaca, "Dynamical Configuration of Measurement Procedures on PDA by Using Measurement Application Repository Server", Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, September 6-8, 2007, pp.120-125.
50. D. Grimaldi, F. Lamonaca, "Analysis of the Synchronization Delay in the Path PC-DSO Co-operating into the W-DMS", Proceedings of the IDAACS 2007-IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, September 6-8, 2007, pp.114-119.
51. D. Grimaldi, F. Lamonaca, Gentile G., "Experimental Analysis of the Delay Introduced by Linux OS in the Communication PC-Measurement Instrument", Proceedings of the IEEE Softcom 2007, Split-Dubrovnik, Croatia, 27-29 September 2007, pp.1-5.
52. D. Grimaldi, F. Lamonaca, "Measurement Techniques to Assess the Time Synchronization in Distributed Systems", Proceedings of the 16th IMEKO TC4 Symposium, Florence, Italy, September 22-24, 2008, pp.480-485.
53. D. Grimaldi, F. Lamonaca, "Delay of the Software Commands in the Path PC-Measurement Instruments", Proceedings of the I2MTC 2008-IEEE International Instrumentation and Measurement Technology Conference, Victoria, Vancouver Island, Canada, May 12-15, 2008, pp.484-488.
54. D. Grimaldi, F. Lamonaca, "Improved Synchronizing Procedure of PDAs to Delivery the Common Sense of the Time to Stand Alone Measurement Instrument", Proceedings of the XIX IMEKO World Congress, Fundamental and Applied Metrology, Lisbon, Portugal, September 6-11, 2009.
55. D. Grimaldi, F. Lamonaca, "Reduction of Polling Cycle to Check Trigger Condition for Instrument Synchronization", Proceedings of the I2MTC 2009-IEEE Intern. Instrum. and Measur. Techn. Conference, Singapore, May 5-7, 2009.
56. Guidelines for synchronization techniques-Accuracy and Availability, 2008.
57. "JCGM 100:2008,GUM 1995 with minor corrections", First edition, JCGM, September 2008, pp.18.

58. "JCGM 100:2008,GUM 1995 with minor corrections",First edition, JCGM, September 2008, pp.28.
59. ISO/IEC Guide 98: Guide to the Expression of Uncertainty in Measurement (GUM), Published by the Intern. Org. for Standardization, 1995.
60. J. H. Hahn, R. Jones, J. Achkar, P. Tuckey, J. M. Pieplu, "Galileo's Timekeeping Infrastructure: Where Do We Go With The External Time Service Provider?", Proceedings of the IEEE International Frequency Control Symposium, joint with the 21st European Frequency and Time Forum, May 29-June 1, 2007, pp.452-457.
61. C. J. Hegarty, E. Chatre, "Evolution of the Global Navigation Satellite System (GNSS)", Proceedings of the IEEE, Vol. 96, No. 12, December 2008, pp.1904-1917.
62. Y. Hirao, K. Tamukai, Y. Kinouchi, H. Yamaguchi, "Synchronized Measurements of Blood Flow Velocity Distributions in Carotid, Brachial and Femoral Arteries, and ECG in Human During Exercise", Proceedings of the First Joint Conf. BMES/EMBS, 1999, vol. 1, 13-16 October, 1999, pp.224.
63. Y. Hirao, T. Kuroda, D. Zhang, Y. Kinouchi, H. Yamaguchi, K. Yoshizaki, "Synchronized Measurements of Maximum Blood Flow Velocities In Carotid, Brachial And Femoral Arteries, and ECG In Human Posture Changes", Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2001, vol. 1, October 25-28, 2001, pp.139-142.
64. R. Hlaváè, M. Lösch, F. Luongo and J. Hahn, "Timing Infrastructure for Galileo System", Proceedings of the 20th EFTF, Braunschweig, Germany, 2006, pp.391-398.
65. K. E. Holbert, G. T. Heydt, H. Ni, "Use of Satellite Technologies for Power System Measurements, Command, and Control", Proceedings of the IEEE 93(5), pp.947-855.
66. R. Horak, "Communication Systems & Networks-Sistemi di Comunicazioni e Reti", Apogeo, 2000, pp.39-48.
67. R. Horak, "Communication Systems & Networks-Sistemi di Comunicazioni e Reti", Apogeo, 2000, pp.470.
68. IEEE std. 1588-2002 "IEEE Standard for a Precise Clock Synchronization Protocol for Network Measurement and Control Systems", published by The Institute of Electrical and Electronics Engineers, New York (USA), November 8, 2002.
69. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE, New York, USA, July 24, 2008, p.177.
70. IEEE 1646-2004, "IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation".
71. H. Jinlun, H. Peicheng, "The Local Restitution of GPS Time at Shanghai Observatory", Proceedings of the International Conference on Microwave and Millimeter Wave Technology Proceedings of ICMMT '98, August 18-20, 1998, pp.467-471.
72. J. Kannisto, T. Vanhatupa, M. Hannikainen, T. D. Hamalainen, "Software and Hardware Prototypes of the IEEE 1588 Precision Time Protocol on Wireless LAN", Proceedings of the 14th IEEE Workshop on Local and Metropolitan Area Networks, September 18, 2005, p.6.
73. E. Kaplan, C. Hegarty, Eds., "Understanding GPS: Principles and Applications, 2nd ed", Norwood, MA, Artech House, 2006.

74. W. J. Klepczynski, "GPS for Precise Time And Time Interval Measurement," in *Global Positioning System: Theory and Applications*, Volume II, B. W. Parkinson and J. J. Spilker, Jr., Eds. Washington, DC: American Institute of Aeronautics and Astronautics, ch. 17, 1996, pp.483-500.
75. Y. Lepage, P. Iarrea, "Unix Bible", IDG Books, Chicago, 2000, pp.70.
76. W. Lewandowski, C. Thomas, "GPS time transfer", *Proceedings of the IEEE* Volume 79, Issue 7, July, 1991, pp.991-1000.
77. W. Lewandowski, J. Danaher, W.J. Klepczynski, "GLONASS Common-View Time Transfer Between North America and Europe and its Comparison with GPS", *Proceedings of the European Frequency and Time Forum, EFTF 96, Tenth (IEE Conf. Publ. 418)*, March 5-7, 1996, pp.388-392.
78. W. Lewandowski, J. Azoubib, W. J. Klepczynski, "GPS: Primary Tool for Time Transfer", *Proceedings of the IEEE*, Volume 87, Issue 1, January, 1999, pp.163-172.
79. R. Lichteneker, "Terrestrial Time Signal Dissemination", *Real-Time Systems*, Vol. 12, Boston, 1997, pp.41-61.
80. W.C. Lindsey, F. Ghazvinian, W.C. Hagmann, K. Dessouky, "Network Synchronization", *Proceedings of the IEEE*, vol. 73, N. 10, October, 1985, pp.1445-1467.
81. M. A. Lombardi, "NIST Time and Frequency Services", NIST Special Publication 432, 2002 Edition, p.29.
82. M. A. Lombardi, "NIST Time and Frequency Services", NIST Special Publication 432, 2002 Edition, p.32.
83. M. A. Lombardi, L.M. Nelson, A. N. Novick, V. S. Zhang "Time and Frequency Measurements Using the Global Positioning System", *The International Journal of Metrology*, July-September, 2001, pp.26-33.
84. J. Lu, H. Leung, "Synchronization: a Fundamental Phenomenon in Complex Dynamical Networks", *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2005, May 23-26, 2005, pp.300-303.
85. R. Lukaszewski, W. Winiecki, "Petri Nets in Measuring Systems Design", *IEEE Transactions on Instrumentation and Measurement*, Volume 57, Issue 5, May, 2008, pp.952-962.
86. P. Mack, F. Capitanescu, M. Glavic, F. Legrand, L. Wehenkel "Application of the Galileo System for a Better Synchronization of Electrical Power Systems", *Proceedings of the IEEE Power Tech Conference*, Lausanne, July, 2007.
87. J. MacKay, "Interfacing Mil Standard Equipment to an IEEE 1588 Enabled Ethernet Network", in NIST publication NISTIR 7192, *Proceedings of the Conference on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, Gaithersburg, Maryland, September 27-29, 2004.
88. J. G. McNeff, "The Global Positioning System", *IEEE Transactions on Microwave Theory and Techniques*, Volume 50, Issue 3, March, 2002, pp.645-652.
89. J.D. Meier, S. Vasireddy, A. Babbar, A. Mackman, "How To: Time Managed Code Using QueryPerformanceCounter and QueryPerformanceFrequency", available at <http://msdn.microsoft.com/en-us/library/ms979201.aspx>.
90. P. Meinhardt, "Time Synchronized End to End Testing Using IRIG-B", *IET Proceedings of the 9th International Conference on Developments in Power System Protection*, March, 17-20 2008, pp.611-614.
91. A.P.S. Meliopoulos, B. Fardanesh, S. Zelingher, G.J. Cokkinides, "Harmonic Measurement System Via Synchronized Measurements", *Proceedings of the*

- IEEE Power Engineering Society Summer Meeting, vol. 2, July 16-20, 2000, pp.1094-1100.
92. D.L. Mills, "Network Time Protocol Version 4 Reference and Implementation Guide", Electrical and Computer Engineering Technical Report 06-06-1, University of Delaware, June 2006, p.83.
  93. D.L. Mills, "Internet Time Synchronization: the Network Time Protocol", Electrical Eng. Dept., Univ. of Delaware, <http://citeseer.nj.nec.com/mills91internet.html>.
  94. P. Misra, P. Enge, Global Positioning, "System: Signals, Measurements, and Performance, 2nd ed.", Lincoln, MA: Ganga-Jamuna, 2006.
  95. M. Mock, R. Frings, E. Nett, S. Trikaliotis, "Continuous Clock Synchronization in Wireless Real-Time Applications", Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems, SRDS-2000, October 16-18, 2000, pp.125-132.
  96. D. S. Mohl, "IEEE 1588 Network Devices", in NIST publication NISTIR 7070, Proceedings of the Workshop on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, September 24, 2003, Gaithersburg, Maryland.
  97. R. A. Nelson, D. D. McCarthy, A. Gifford, T. R. Bartholomew, "Timescales for the GPS and Galileo" Proceedings of the 18th European Frequency and Time Forum, EFTF 2004, April 5-7, 2004, pp.182-186.
  98. <http://forums.ni.com/ni/board/message?board.id=140&message.id=16266>.
  99. M.M. Nordman, W.E. Kozlowski, O. Vahamaki, "A Method for Synchronizing Low Cost Energy Aware Sensors Used in Industrial Process Monitoring", Proceedings of the 27th Annual Conf. of the IEEE Ind. Electr. Soc., IECON '01, vol. 1, November 29-December 2, 2001, pp.100-106.
  100. K. O'Donoghue, et al., "IEEE 1588 in Navy Next Generation Time Sensitive Applications", Proceedings of the Conference on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, October 2-4, 2006, Gaithersburg, Maryland.
  101. O. Orpen, H. Zwaan "Dual Frequency DGPS Service for Combating Ionospheric Interference", The Journal of Navigation Vol. 54, Nr. 1, pp.29,36.
  102. S. Palchadhuri, A.K. Saha, D.B. Johns, "Adaptive Clock Synchronization in Sensor Networks", Proceedings of the Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004, April 26-27, 2004, pp.340-348.
  103. B. Parkinson, S. Gilbert, "NAVSTAR: Global Positioning System-Ten Years Later", Proceedings of the IEEE, October, 1983.
  104. B. Parkinson and J. J. Spilker, Jr., Eds., "Global Positioning System: Theory and Applications" Washington, D.C.: AIAA, 1996, vol. I.
  105. D. Perino, Statement by the Press Secretary, White House, Office of the Press Secretary, Washington, D.C., September 18, 2007.
  106. H. Ping, Bing-Fa Zu, L. Yang Liu, "GPS Timer Transfer Smooth Based on Kalman Filter", NAVSTAR: Global Positioning System-Ten Years Later 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM'08, October 12-14, 2008, pp.1-4.
  107. E. Powers, J. Hahn, "GPS and Galileo UTC Time Distribution", Proceedings of the 18th European Congress on Frequency and Time Forum, 2004. EFTF 2004, April 5-7 2004, pp.484-488.
  108. "RabbitCore RCM4400W C-Programmable Wi-Fi Core Module", OEM User's Manual.

109. "RabbitCore RCM4400W OEM User's Manual", Rabbit Semiconductor Inc., USA, 2007.
110. G. Racciu, P. Mantegazza, "RTAI User Manual" 3.4-October 2006-rev.03.
111. T. Radil, P.M. Ramos, A. Cruz Serra, "DSP Based Portable Impedance Measurement Instrument Using Sine-Fitting Algorithms", Proceedings of the IEEE IMTC/05, Ottawa (Canada), vol. 2, May, 2005, pp.1018-1022.
112. L. Ramadoss, J.Y. Hung, "A study on Universal Serial Bus Latency in a Real-Time Control System", Proceedings of the 34th Annual Conference of IEEE Industrial Electronics, November 10-13, 2008, pp.67-72.
113. S. Rodrigues, "Combining Synchronous Ethernet and IEEE 1588 for use in Telecom", Proceedings of the Conference on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Gaithersburg, Maryland, October 2-4, 2006.
114. S. Rodrigues, "Standards Update-IEEE 1588", The 6th Time & Synchronisation in Telecoms Conference, November 4-6, 2008.
115. G. Racciu, P. Mantegazza, "RTAI User Manual 3.4-October 2006-rev 0.3".
116. J. Kiszka, B. Wagner, Y. Zhang, J. Broenink, "RTnet-A Flexible Hard Real-Time Networking Framework", Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, Catania, Italy, September 19-22, 2005.
117. H. Saitoh, "GPS Synchronized Measurement Applications in Japan", Proceedings of the Asia Pacific. IEEE/PES Transmission and Distribution Conference and Exhibition 2002, vol. 1, October 6-10, 2002, pp.494-499.
118. H. Saitoh, K. Miura, O. Ishiok, H. Sato, J. Toyoda, "On-line Modal Analysis Based on Synchronized Measurement Technology", Proceedings of the PowerCon 2002. Intern. Conference on Power System Technology, vol. 2, October 13-17, 2002, pp.817-822.
119. J. Sanchez, F. Morilla, S. Dormido, J. Aranda, P. Ruiperez, "Virtual and Remote Control Labs Using Java: a Qualitative Approach", IEEE Control Systems Magazine, Vol. 22, No. 2, 2002, pp.8-20.
120. J. Schoukens, F. Louage, Y. Rolain, "Study of the Influence of Clock Instabilities in synchronized Data Acquisition Systems", IEEE Transactions on Instrumentation and Measurement, vol. 45, N. 2, April, 1996, pp.601-604.
121. Séroul, R. "Lagrange Interpolation." in Programming for Mathematicians. Berlin: Springer-Verlag, 2000, pp.269-273.
122. M. E. Shepard, D. G. Fowley, R. L. Jackson, D. B. King, "Implementation of IEEE Sdc.-1588 in a Networked I/O Node", in NIST publication NISTIR 7070, Proceedings of the Workshop on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Gaithersburg, Maryland, September 24, 2003.
123. A. Silberschatz, P. B. Galvin, G. Gagne, "Sistemi Operativi, Concetti ed esempi", 6° ed., Pearson Education Italia, 2002, pp.155-192.
124. V.V. Spiridonov, "Inmarsat Systems and Services", Proceedings of International Conference on Satellite Communications, Volume 1, October 18-21, 1994, pp.45-52.
125. A.S. Tanenbaum, "Modern Operating Systems", second edition, Prentice Hall, 2001.
126. M. J. Teener, "Ethernet Bridging and Ethernet AV<sup>TM</sup>", <http://www3.ietf.org/proceedings/06nov/slides/tsvarea-1.pdf>.

127. <http://forums.ni.com/ni/board/message?board.id=140&message.id=16266>.
128. <http://cnt.rm.ingv.it/>.
129. D. Tonks, "IEEE 1588 in Telecommunications and Field Trial Results", Proceedings of the 2005 Conference on IEEE-1588, Zurich, Switzerland, October 10-12, 2005.
130. J. C. Tournier, X. Yin, "Improving Reliability of IEEE1588 in Electric Substation Automation", Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2008.
131. J.C. Tournier, O. Goerlitz, "Strategies to Secure the IEEE 1588 Protocol in Digital Substation Automation", Proceedings of the 4th International Conference on Critical Infrastructures, March 27-April 30, 2009, pp.1-8.
132. H. Ukai, K. Nakamura, S. Nishigaki, Y. Ohta, N. Matsui, "Advanced Synchronized Measurement System of Harmonics Using DSP and GPS", Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society, vol. 2, October 22-28, 2000, pp.748-753.
133. H. Ukai, K. Nakamura, N. Matsui, "DSP-and GPS-based Synchronized Measurement System of Harmonics in Wide-Area Distribution System", IEEE Transactions on Industrial Electronics, vol. 50, N. 6, December, 2003, pp.1159-1164.
134. D. Vook, B. Hamilton, A. Fernandez, J. Burch, V. Srikantam, "Update on High Precision Time Synchronization", Proceedings of the Conference on IEEE-1588, Zurich, Switzerland, October 10-12, 2005.
135. J. Weare, "Kernel Comparison: Linux (2.6.22) versus Windows (Vista), Clock Hardware Clock" consultable on <http://widefox.pbwiki.com/Kernel%20Comparison%20Linux%20vs%20Windows> at August 08, 2008.
136. R.E. Wilson, P.S. Sterlina "Verification of Measured Transmission System Phase Angles", IEEE Transaction on Power Delivery, vol.11, N.4, October, 1996, pp.1743-1747.
137. W. Winiecki, M. Karkowski, "A New Java-based Software Environment for Distributed Measuring Systems Design", IEEE Transactions on Instrumentation and Measurement, Vol.51, No.6, 2002, pp.1340-1346.
138. J. R. Wright, "GPS Composite Clock Analysis", Proceedings of the IEEE International Frequency Control Symposium, joint with the 21st European Frequency and Time Forum, 2007.
139. P. Gerum, "Xenomai-Implementing a RTOS Emulation Framework on GNU/Linux", April 2004.
140. N.Ya'acob, M. Abdullah, M. Ismail, "Multipath Mitigation of Global Positioning System (GPS) Signal Using Wavelet Technique", Proceedings of the International Conference on Digital Image Processing, March 7-9, 2009, pp.142-146.
141. E. Yanmaz, "Mitigating Effects of Node Failure via Mobility in Wireless Networks", Proceedings of the 4th International Symposium on Wireless Pervasive Computing-ISWPC, 2009, pp.1-5.
142. P. Yaou, "Windows CE 3.0: Enhanced Real-Time Features Provide Sophisticated Thread Handling", issue of MSDN Magazine, November, 2000.
143. R. Zanello, M. Mascarello, P. Tavella, L. Galleani, E. Detoma, A. Bellotti, "The Galileo Precise Timing Facility", Proceedings of the IEEE International Frequency Control Symposium, joint with the 21st European Frequency and Time Forum, May 29-June 1, 2007, pp.458-462.

144. R. Zanello, M. Blanchi, C. Piras, E. Detoma, P. Capetti, A. Bellotti, D. Villabruna, "Implementing the Galileo Precise Timing Facility", Proceedings of the IEEE International Frequency Control Symposium, joint with the 22nd European Frequency and Time forum, April 20-24, 2009, pp.647-652.
145. S. Zelingher, B. Fardanesh, E. Uzunovic, A.P.S. Meliopoulos, G. Cokkinides, "Harmonic Monitoring System Via GPS-Synchronized Measurements-Update and New Developments", Proceedings of the IEEE Power Engineering Society General Meeting, June 18-22, 2006, p.7.
146. F. Zhang, G.Y. Deng, "Probabilistic time synchronization in wireless sensor networks", Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005, vol. 2, September 23-26, 2005, pp.980-984.