Creating Real-Time operation System Based on xPC Target Kernel

Abdullah A. Mohamed, Dia M. Ali

Abstract— Real-Time system is the base for many applications today. Whatever the designed system is a high accuracy and flexibly, it still not efficient unless be a real-time with the overall system. In this paper a real-time Operation System (OS) is created based on xPC Target Kernel. xPC gives the ability to convert a designed model to a real-time OS. Two models are built, Target-Host transmission using UDP and Spectrum Analyzer, and implemented separately on the created OS. The first model to test the ability of the created OS when dealing with network systems, the second to test the ability of dealing with High Speed Systems and Analyzing units. There result shows that the created OS is very efficient and give short average Task Execution Time (TET) equal to (0.636 micro sec) for the first model and (0.3117 micro sec) for the second model and it shows that it can easily communicate with other systems.

Index Terms— Personal Computer (PC), Task Execution Time (TET), Snapshot from Simulink-Matlab R2010 (SSMR10).

I. INTRODUCTION

Today the speed becomes the more important factor for any system, computer networks, control units and counting systems. Therefore, if the designed project has a high accuracy, simplicity or flexibility but unable to deal with the all over system speed, it will be the bottleneck and it will not be efficient system. Designing a project and implementing it on none real-time Operating System (OS), Windows, Mac or Linux not give solution for the execution speed because the CPU will be interrupted from many tasks that not useful for the main aim for the designed project like mouse, keyboard, sound card and unused ports. The right solution is to convert the designed project to a real-time OS, it is a magic, xPC is the magic tool that gives the ability to adopting the Kernel and convert the designed project to a custom lite OS. This facility increases the speed of the execution and makes it a Real-time system. xPC overcome the real-time and it gives a True time operation time. It can be used in many fields which required high-speed execution as control systems [1], [3] and computer network systems [4].

II. REAL-TIME VS. TRUE-TIME

Real-time is a more flexible word. If there is a network work with a speed 1 Kbyte/sec and the designed system can give an output by speed less than 1 Kbyte/sec, it will be a real-time system with respect to that network, so it depend on the speed of the network and there aren’t a limited speed of execution to be a standard for real time that can according to it can be decided that the system is a real-time or not [2].

True-time is a word that less than confusion from the previous one. It take a virtual environment as the real-world for the network environment and consider the designed system applied on a PC connected with that real-world, then it make a ratio between the CPU clock of the a assumed PC, virtual environment and the CPU clock of the devices that used in the designing. True-time is more realty from Real-Time and it give the true result in testing and implementing [1].

Revised Manuscript Received on 30 September 2013.
* Correspondence Author

Abdullah A. Mohamed, Communications Department, Mosul University/ College of Electronics Engineering, Mosul, Iraq.
Dia M. Ali, Communications Department, Mosul University/ College of Electronics Engineering, Mosul, Iraq.

III. XPC

xPC is one of the most powerful tools library of Simulink Toolbox. It is a magic tool consist of two parts the Target product and the Host product. It product is a best solution for rapid and fast control prototyping, testing, Hardware In the Loop (HIL) and deploying real-time systems using standard PC hardware. Target PC is separated from a Host PC. Host PC is used for controlling and monitoring the work of the project in the Target PC. The xPC software environment includes many features that help the designer prototype, test and deploy real-time systems [7]. It contain many blocks that can easily used in the designing Fig (1). These blocks classified to many categories Ethernet, IP carrier and many more others. The general Simulink blocks are also supported like signal generators, scopes, etc.

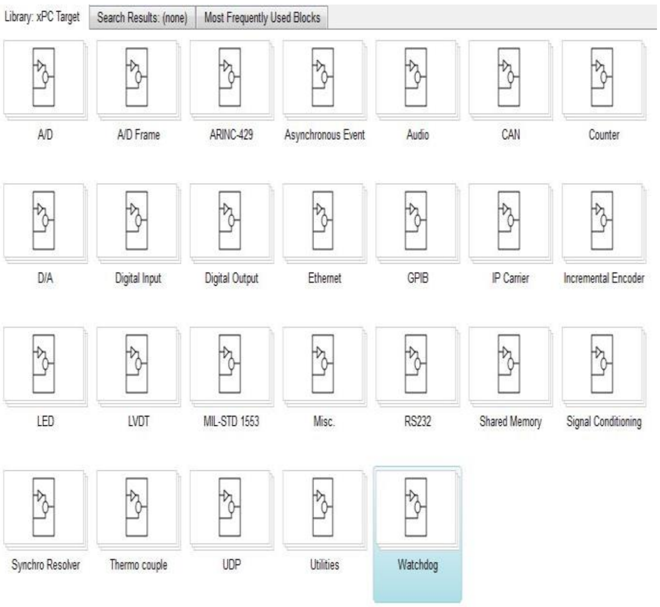
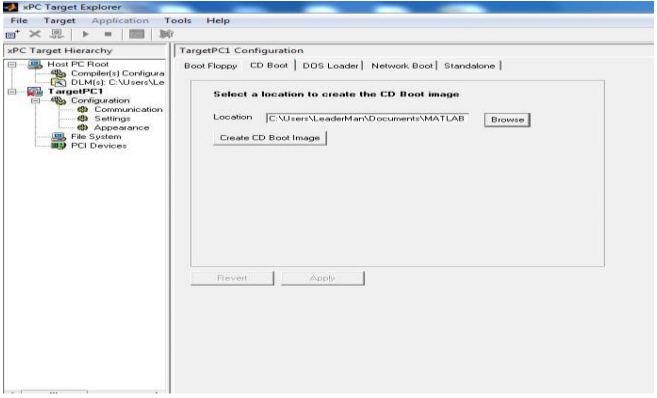


Fig. 1 xPC Target Library. (SSMR10).

*Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved*

IV. CREATING REAL-TIME OPERATING SYSTEM

xPC use xPC Target kernel as a base to create a custom operating system, for that it does not require Linux, DOS, Windows or any another OS on the target PC. Instead, the target PC is booted by boot media which includes the xPC Target kernel [5]. xPC Target software require C/C++ compiler, Real-Time Workshop, State flow Coder (optional), Real-Time Workshop Embedded Coder (optional) software to convert the designed model to an executable code. This code is downloaded from the host PC to the target PC that running the xPC Target kernel, then it can be run in real-time. Additionally, I/O blocks can be added to the designed model to connect and communicate with other hardware. To simplify what mentioned above, it is subdivided to steps.



Select a location to create the CD Boot image

Location: C:\Users\LeaderMan\Documents\MATLAB

Fig. 2 xPC-Explorer. (SSMR10).

- **Step 1**: Create and Burn Kernel on Boot Media

xPC provides xPC-Explorer tool that through it configurations can be set to create the right adopted xPC Kernel. This tool execute by putting "xpcexpr" in the Matlab command window, fig(2) shows xPC-Explorer, under the TargetPC1 Configuration there are number of configuration tasks Boot Floppy, CD Boot and others. By choosing "CD Boot" option, it will ask to define the location to create the CD Boot image in it. Through "Browser", option the location set in any folder or directly on the writable disk CD/DVD. By click on "Create CD Boot image", the image (xPC Target kernel) will created automatically, and then click on "Apply" the disk will start burning.

- **Step-2**: Make Target-Host Communication

After creating boot image, the Target PC can easily boot from it. To connect Host PC with Target PC, some configuration is required. xPC allow making Host-Target connection through two communication methods RS232 and TCP/IP. Fig (3) shows xPC-Explorer, under the "Communication" there are three fields.

A. *Communication protocol:*

through "Host Target Communication "the protocol can be set as RS232 or TCP/IP.

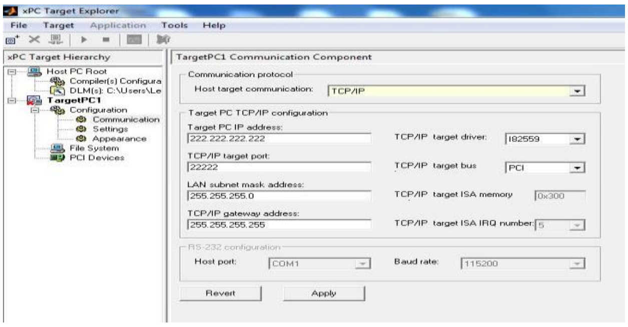
B. *Target PC TCP/IP Configuration:*

Target IP address can be set through "Target PC IP address". Target driver can be set through "TCP/IP Target driver", xPC support limited vender types of Ethernet cards [8]. All the other fields can be left as default.

C. *RS232 Configuration:*

if the Host PC has more than one COM, the COM number must be define and the baud rate also.

By click "apply", all the setting will be saved and the real-time system on the Target PC is ready to receive the downloaded model from the Host PC.



TargetPC1 Communication Component

Communication protocol
Host target communication: TCP/IP

Target PC TCP/IP configuration
Target PC IP address: 222.222.222.222
TCP/IP target driver: I82559
TCP/IP target port: 22222
TCP/IP target bus: PCI
LAN subnet mask address: 255.255.255.0
TCP/IP target ISA memory: 0x300
TCP/IP gateway address: 255.255.255.255
TCP/IP target ISA IRQ number: 5

RS-232 configuration
Host port: COM1
Baud rate: 115200

Fig. 3xPC communication configurations. (SSMR10)

V. BUILDING TWO MODELS AND TESTING IT ON THE CREATED OS

To test the features of the created OS, two models are built and applied separately on it.

A. *Target-Host transmission using UDP*

Bytes stream generator is designed Fig (4). Three PCs and network switch are used to implement the model as shown in Fig (5). When boot the Target PC by the created OS, it will appear as shown in Fig (6). It gives the minimum required details that the Target PC (x86 2.2GHz Core 2 Duel CPU and 2GB RAM) is connected with the Host PC through TCP/IP (Ethernet). When opening the model in the Host PC, it can easily controlled through the xPC-Explorer Fig (7). By clicking "connect" on "TargetPC1", the states will become connected "yes" and then it will allow downloading the model to the Kernel of the Target PC's OS. When then start run the model, Target PC will execute it Fig (8) and it will give the max., min. and the average Task Execution Time (TET) " TET is the time in seconds to complete calculations for the model equations and post outputs during each sample interval [6]". All the information that display in Target PC Monitor can be monitored and controlled in the Host PC through xPC-Explorer. Target PC generate a UDP packets with destination IP of the third PC and send it to the Switch that forward it to the third PC which revived it successfully.

B. *Spectrum Analyzer*

By using the same configuration in model "A", model "B" is built Fig(9) and by using the same "A" steps, Target PC executed the model and display the frequency component that generated in the model as shown in Fig(10).

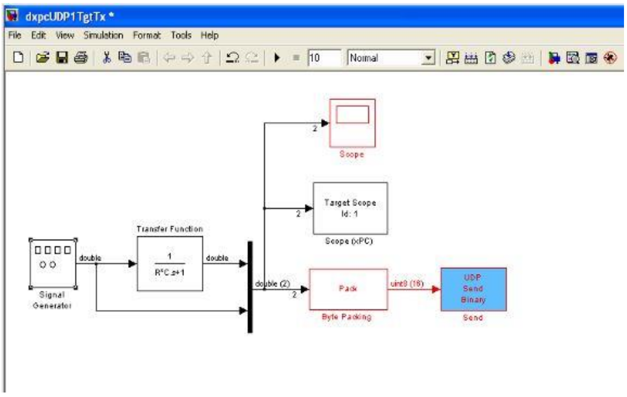


Fig. 4 Model one. (SSMR10).

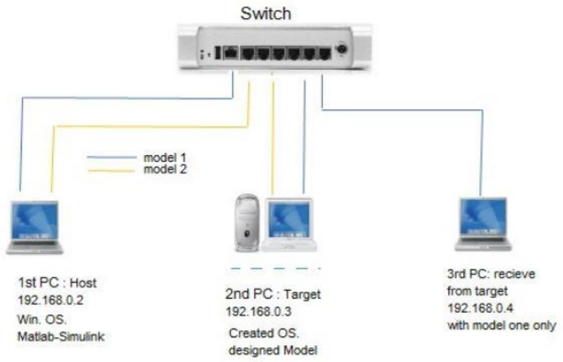
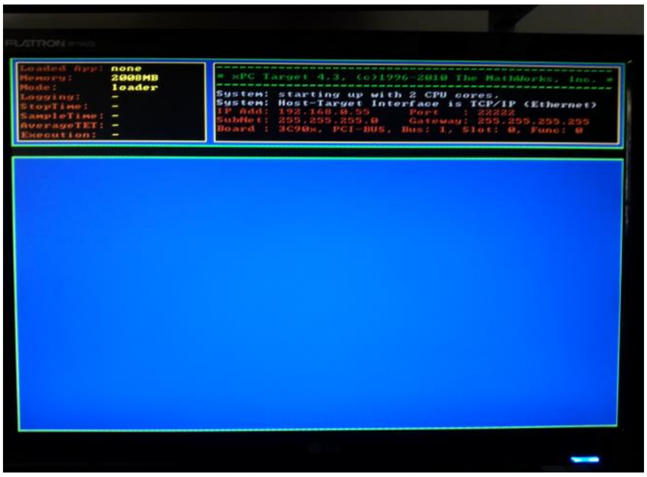


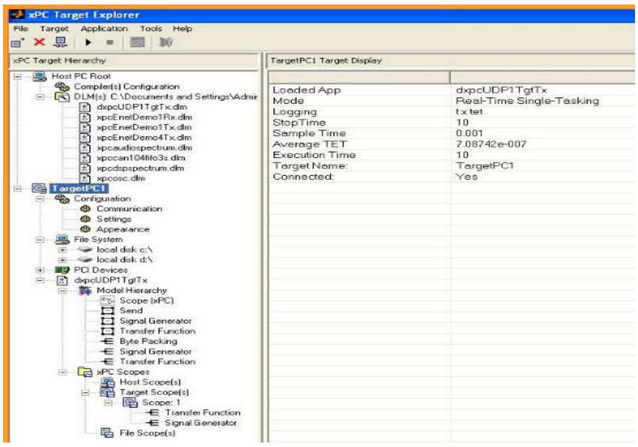
Fig. 5 Model 1, 2 topology



```
Loaded App:   none
Memory:       2008MB
Mode:         loader
Logging:      -
StopTime:     -
SampleTime:   -
AverageTET:   -
Execution:    -
```

```
* xPC Target 4.3, (c)1996-2010 The MathWorks, Inc. *
System: starting up with 2 CPU cores.
System: Host-Target Interface is TCP/IP (Ethernet)
IP Add: 192.168.0.55        Port    : 22222
SubNet: 255.255.255.0       Gateway: 255.255.255.255
Board : 3C90x, PCI-BUS, Bus: 1, Slot: 0, Func: 0
```

Fig. 6 The created real-time OS at booting



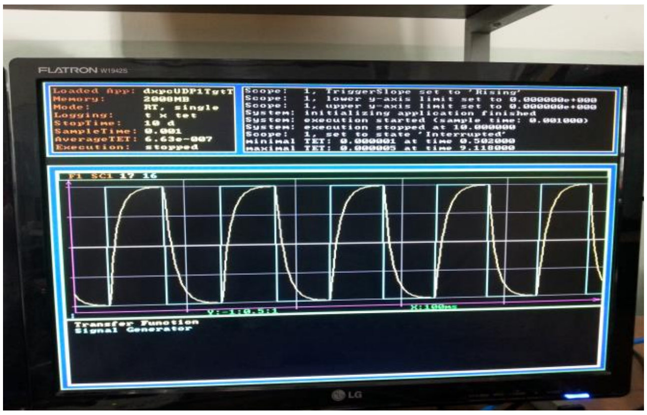
xPC Target Hierarchy

- Host PC Root
  - Compiler(s) Configuration
  - DLM(s): C:\Documents and Settings\Admir
    - dxpcUDP1TgtTx.dlm
    - xpcEnetDemo1Rx.dlm
    - xpcEnetDemo1Tx.dlm
    - xpcEnetDemo4Tx.dlm
    - xpcaudiospectrum.dlm
    - xpccan104fifo3s.dlm
    - xpcdspspectrum.dlm
    - xpcosc.dlm
- TargetPC1
  - Configuration
    - Communication
    - Settings
    - Appearance
  - File System
    - local disk c:\
    - local disk d:\
  - PCI Devices
  - dxpcUDP1TgtTx
    - Model Hierarchy
      - Scope (xPC)
      - Send
      - Signal Generator
      - Transfer Function
      - Byte Packing
      - Signal Generator
      - Transfer Function
    - xPC Scopes
      - Host Scope(s)
      - Target Scope(s)
        - Scope: 1
          - Transfer Function
          - Signal Generator
      - File Scope(s)

TargetPC1 Target Display

Loaded App	dxpcUDP1TgtTx
Mode	Real-Time Single-Tasking
Logging	t x tet
StopTime	10
Sample Time	0.001
Average TET	7.08742e-007
Execution Time	10
Target Name:	TargetPC1
Connected:	Yes

Fig. 7 control window in xPC-Explor. Model 1 (SSMR10).



```
Loaded App:   dxpcUDP1TgtT
Memory:       2008MB
Mode:         RT, single
Logging:      t x tet
StopTime:     10 d
SampleTime:   0.001
AverageTET:   6.63e-007
Execution:    stopped
```

```
Scope:   1, TriggerSlope set to 'Rising'
Scope:   1, lower y-axis limit set to 0.000000e+000
Scope:   1, upper y-axis limit set to 0.000000e+000
System:  initializing application finished
System:  execution started (sample time: 0.001000)
System:  execution stopped at 10.000000
Scope:   1, set to state 'Interrupted'
minimal  TET: 0.000001 at time 0.502000
maximal  TET: 0.000005 at time 9.118000
```

```
Transfer Function
Signal Generator
```

Fig. 8 The execution of Model 1 in the created OS

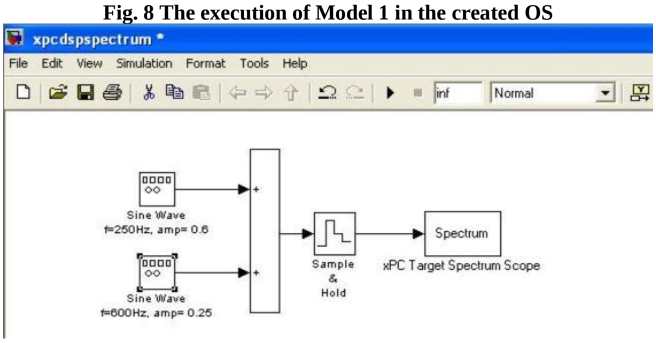
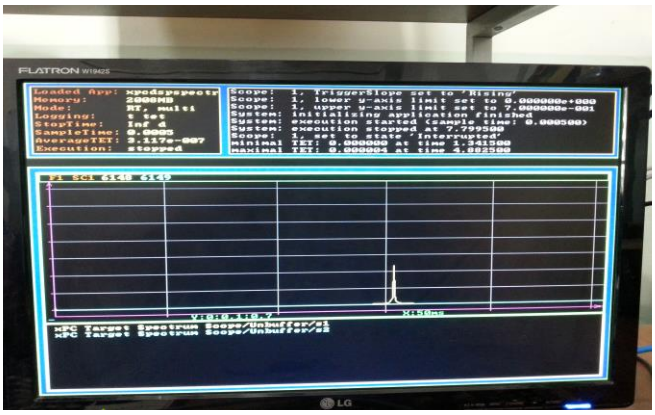


Fig. 9 Model 2 (SSMR10).



```
Loaded App:   xpcdspspectr
Memory:       2008MB
Mode:         RT, multi
Logging:      t tet
StopTime:     Inf d
SampleTime:   0.0005
AverageTET:   3.117e-007
Execution:    stopped
```

```
Scope:   1, TriggerSlope set to 'Rising'
Scope:   1, lower y-axis limit set to 0.000000e+000
Scope:   1, upper y-axis limit set to 7.000000e-001
System:  initializing application finished
System:  execution started (sample time: 0.000500)
System:  execution stopped at 7.799500
Scope:   1, set to state 'Interrupted'
minimal  TET: 0.000000 at time 1.341500
maximal  TET: 0.000004 at time 4.882500
```

```
xPC Target Spectrum Scope/Unbuffer/s1
xPC Target Spectrum Scope/Unbuffer/s2
```

Fig. 10 The execution of model 2 in the created OS.

VI. RESULTS AND CONCLUSIONS

xPC is a very efficient tool to create a real time system that can be used in many fields Networks, Control Unites and High Speed Projects. The created OS is tested with two models Target-Host transmission using UDP and Spectrum Analyzer. The first model is chosen to see the OS how it can deal with the network systems and check the ability to communicate with other external systems. The second model is chosen to see the ability of dealing with Control units and Analyzing systems. Table 1 shows that the OS is very efficient and it gives TET very less than the same models when it run in side the Matlab-Simulink with the same hardware PC, this difference came from that the model when run in side the Simulink it take a little care from the CPU that is because the CPU process the Windows tasks and it run the Simulink inside the Windows and the model inside the Simulink.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

These series of levels interrupt the CPU process, but when the model run inside the created OS, the CPU will not have any work to do just the model and it will take all the CPU resources. About ten iterations are done for each case and the records are summered in Table 1.

Table 1 The Results.

States			Min. TET	Max. TET	Avg. TET
Run for first time	Run in Simulink	Model -1	1.2 sec.	3.6 sec.	2.6 sec.
		Model -2	0.9 sec.	2.3 sec.	1.3 sec.
	Run in created OS.	Model -1	0.1 micro-sec.	5 micro-sec.	0.663 micro-sec.
		Model -2	0.01 micro-sec.	0.4 micro-sec.	0.311 micro-sec.
Run after the first time	Run in Simulink	Model -1	0.1 sec.	0.3 sec.	0.12 sec.
		Model -2	0.08 sec.	0.12 sec.	0.9 sec.
	Run in created OS.	Model -1	0.1 micro-sec.	5 micro-sec.	0.663 micro-sec.
		Model-2	0.01 micro-sec.	0.4 micro-sec.	0.311 micro-sec.

ACKNOWLEDGMENT

We would like to thank each one who helped us to complete this work. Special thanks for:
Dr. Majid Dhirar - Department of Computer Lecturer Umar Farooq - Department of Communication
Lecturer Ramiz - Department of Communications
Lecturer Mohamed Samir - Department of Communications

REFERENCES

1. Anton Cervin, “Integrated Control and Real-Time Scheduling”, PhD. Thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, pp. 139-161, April 2003.
2. Stefan Molyneux, “Real-Time Relationships-The Logic of Love", Freedomain Library, Volume 4 Version 1.0 Extended Edition, pp. 205-230, January 2008, www.freedomainradio.com.
3. Emeka Eyisi, Jia Bai, Derek Riley, Jiannian Weng, Wei Yan, Yuan Xue, Xenofon Koutsoukos, Janos Sztipanovits, “An integrated modeling and simulation tool for networked control systems”, Simulation Modelling Practice and Theory 27 (2012), Elsevier.
4. Abdullah A. Mohamed, “Designing of Intrusion Detection System Based on Image Block Matching”, International Journal of Computer and Communication Engineering, Vol. 2, No. 5, September 2013.
5. The MathWorks Inc., “xPC Target User’s Guide for Use with Real-Time Workshop”, The MathWorks Inc., pp. 85-92, 2000.
6. The MathWorks Inc., “xPC Target Getting Started Guide”, The MathWorks Inc., pp. 10-31, 2010.
7. The MathWorks Inc., “xPC Target”, The MathWorks Inc., 2012.
8. The MathWorks Inc., “xPC_Target_Supported_Ethernet_Chipsets”, The MathWorks Inc., 2010.

AUTHOR PROFILE

Abdullah Abdulaziz Mohamed was born in Mosul-Iraq in 1988. He received his engineering degree in Communication Science from the University of Mosul in 2011. He is a student in master study of network security since 2011 until now. His published papers are “Designing of Intrusion Detection System Based on Image Block Matching” and “Challenges of Designing and Implementing Intrusion Detection System”

Dia M. Ali:recieved his BSC in Elec. Comm. Eng. From Mosul university in 1992 (Mosul/ Iraq). In 1998 he gets MSC in Elec. Comm. Eng. Mosul University (Mosul / Iraq) and 2007-he recieved a PhD in Comm. Eng. (Network) from Mosul university (Mosul/ Iraq). From 1993 to 2001 he worked at R&D Center Mosul / Iraq . Since 2007 join university of Mosul collage of Engineering as lecture. He interesting in Network Modeling and simulation specially in (OPNET MODELER), Network Security, Mobile Network planning, Antenna modeling and System.