



FENERBAHÇE ÜNİVERSİTESİ

Petalinux Tabanlı CORDIC HW/SW CoDesign Projesi

Evrım Arda Kalafat, Arda Alhan, Cüneyt Balcı

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: evrim.kalafat@stu.fbu.edu.tr, arda.alhan@stu.fbu.edu.tr ,
cuneyt.balci@stu.fbu.edu.tr

Proje özeti: Proje Petalinux ortamından UART arayüzü üzerinden yakalanan iki sayı girişi verilip, beklenen çıktının aynı sonucu elde edilip edilmediği kontrol edilecektir. Sonuç UART arayüzü üzerinden PC'e gönderilecektir.

Anahtar Kelimeler: FPGA, CPU, SystemVerilog, Petalinux, UART.

Abstract: Within the scope of this project, we are going to design ALU and Instruction decoder blocks of a RISC-V processor by using the basic SystemVerilog language features.

Keywords: FPGA, CPU, RISC-V, SystemVerilog, RTL.

I. Giriş

Donanım hızlandırıcı olarak Xilinx'in CORDIC (COordinate Rotation DIgital Computer) IP'si kullanıldığı ve işlemcinin hesap yükünü donanım hızlandırıcıya aktardığı bir proje yapılacaktır. Bu donanım tasarımı PL tarafında hazırlandıktan sonra Petalinux ortamından veriler aktarılacaktır.

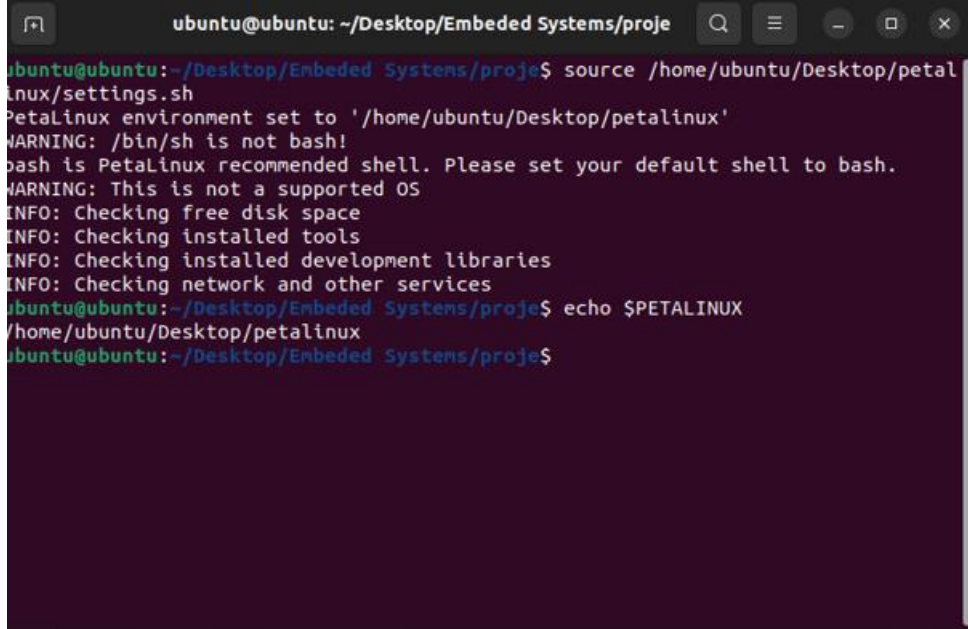
II. Sistem Mimarisi

Xilinx Vivado Design Suite: Xilinx Vivado Design Suite, FPGA geliştirme kartları üzerinde çalışmalar yapmak için gerekli olan tasarımı oluşturmak için kullanılmaktadır. Verilog, VHDL vb.. donanım tasarım dillerini alarak, FPGA'e konfigüre edilebilecek (Xilinx firması FPGA'leri için .bit uzantılı dosyalar) tasarım dosyasını oluşturur.

Visual Studio Community: Proje kapsamında, ödev ve LAB'larda kullanılan Microsoft'un derleyicisi olan Visual Studio Community kullanılacaktır.

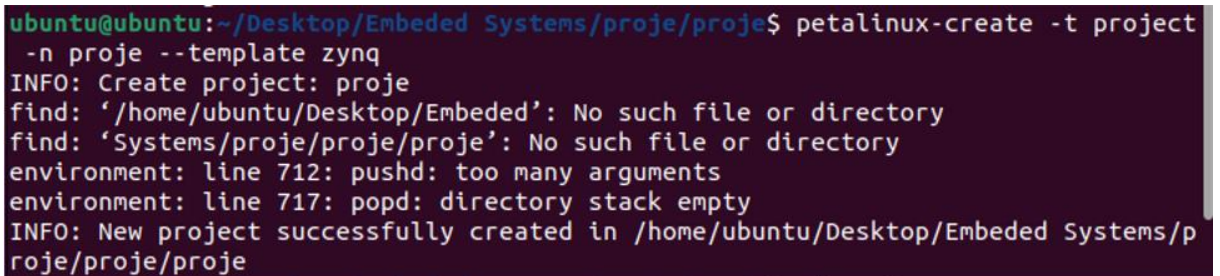
III. Kullanılan Yazılım

- Vivado’da dizaynımızı oluşturuyoruz ve ardından UART arayüzü ayarlamalarını yaptık. Burada kurduğumuz petalinux’ü çalıştırdık ve aktif ettik. Ardından da echo komutu ile doğru çalışıp çalışmadığını test ettik.



```
ubuntu@ubuntu: ~/Desktop/Embedded Systems/proje
ubuntu@ubuntu:~/Desktop/Embedded Systems/proje$ source /home/ubuntu/Desktop/petal
linux/settings.sh
PetaLinux environment set to '/home/ubuntu/Desktop/petalinux'
WARNING: /bin/sh is not bash!
bash is PetaLinux recommended shell. Please set your default shell to bash.
WARNING: This is not a supported OS
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
ubuntu@ubuntu:~/Desktop/Embedded Systems/proje$ echo $PETALINUX
/home/ubuntu/Desktop/petalinux
ubuntu@ubuntu:~/Desktop/Embedded Systems/proje$
```

- Sonrasında proje taslağını oluşturup, içine girdik ve sonrasında da get-hw-description argümanı ile projeyi konfigüre ettik.



```
ubuntu@ubuntu:~/Desktop/Embedded Systems/proje/proje$ petalinux-create -t project
-n proje --template zynq
INFO: Create project: proje
find: '/home/ubuntu/Desktop/Embedded': No such file or directory
find: 'Systems/proje/proje/proje': No such file or directory
environment: line 712: pushd: too many arguments
environment: line 717: popd: directory stack empty
INFO: New project successfully created in /home/ubuntu/Desktop/Embedded Systems/p
roje/proje/proje
```

```
ubuntu@ubuntu: ~/Desktop/Embedded Systems/proje/proje/proje$ petalinux-config --get-hw-descript
ion ../../
[INFO] Sourcing buildtools
INFO: Getting hardware description...
INFO: Renaming design_1_wrapper.xsa to system.xsa
[INFO] Generating Kconfig for project
[INFO] Menuconfig project

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Sourcing buildtools extended
[INFO] Extracting yocto SDK to components/yocto. This may take time!
[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Adding user layers
[INFO] Generating workspace directory
ubuntu@ubuntu:~/Desktop/Embedded Systems/proje/proje/proje$
```

- Sonrasında ethernetten veri yakalayacağımız ve bunu C kodlarıyla yapacağımız için de içeride compiler'lara ihtiyacımız olacak. Bu yüzden de 'petalinux config -c rootfs' komutu ile içeriye compiler paketi koyulmasını sağladık.

```
ubuntu@ubuntu:~/Desktop/Embedded Systems/proje/proje/proje$ petalinux-config -c rootfs
[INFO] Sourcing buildtools
[INFO] Silentconfig project
[INFO] Generating kconfig for Rootfs
[INFO] Menuconfig rootfs

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Generating plnxtool conf
[INFO] Successfully configured rootfs
```

- 'petalinux-config -c kernel' komutu ile kernel için gerekli olan ayarların yapıldığı kısımdan kernel'ı modifiye ettik.


```

ubuntu@ubuntu:~/Desktop/Embedded_Systems/proje/proje/proje$ petalinux-config -c kernel
[INFO] Sourcing buildtools
[INFO] Silentconfig project
[INFO] Sourcing buildtools extended
[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Generating workspace directory
[INFO] Configuring: kernel
[INFO] bitbake virtual/kernel -c cleansstate
NOTE: Started PRServer with DBfile: /home/ubuntu/Desktop/Embedded_Systems/proje/proje/build/cache/prserv.sqlite3, IP: 127.0.0.1, PORT: 35937, PID: 12446
WARNING: Host distribution "ubuntu-22.04" has not been validated with this version of the build system; you may possibly experience unexpected failures. It is recommended that you use a tested distribution.
Loading cache: 100% | ETA: --:--:--
Loaded 0 entries from dependency cache.
Parsing recipes: 100% |#####| Time: 0:04:42
Parsing of 3476 .bb files complete (0 cached, 3476 parsed). 5133 targets, 268 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

```

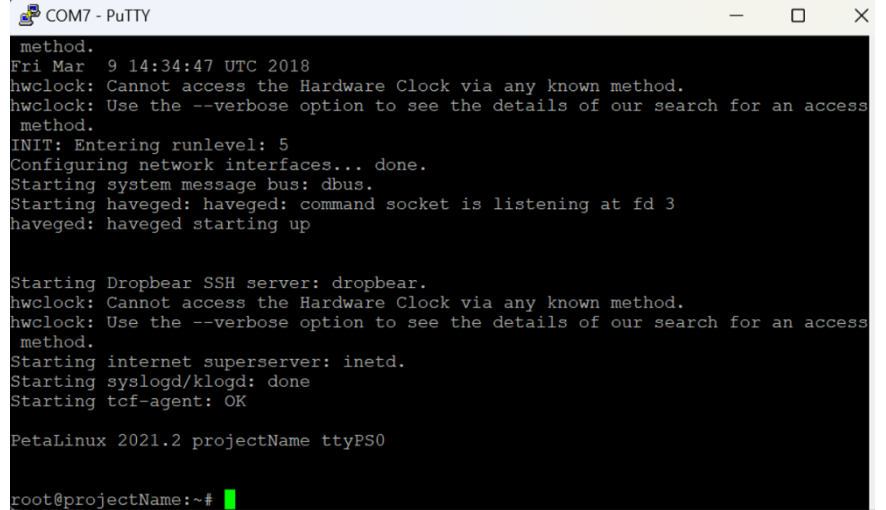
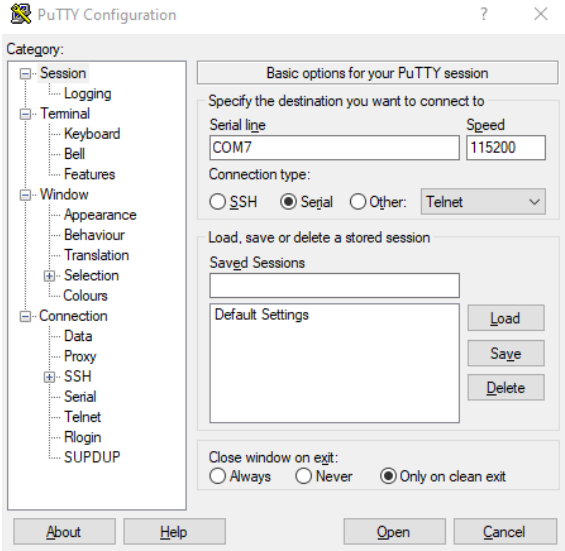
- Sonrasında ise ‘petalinux-build’ komutu ile paketi compile ettik.

```

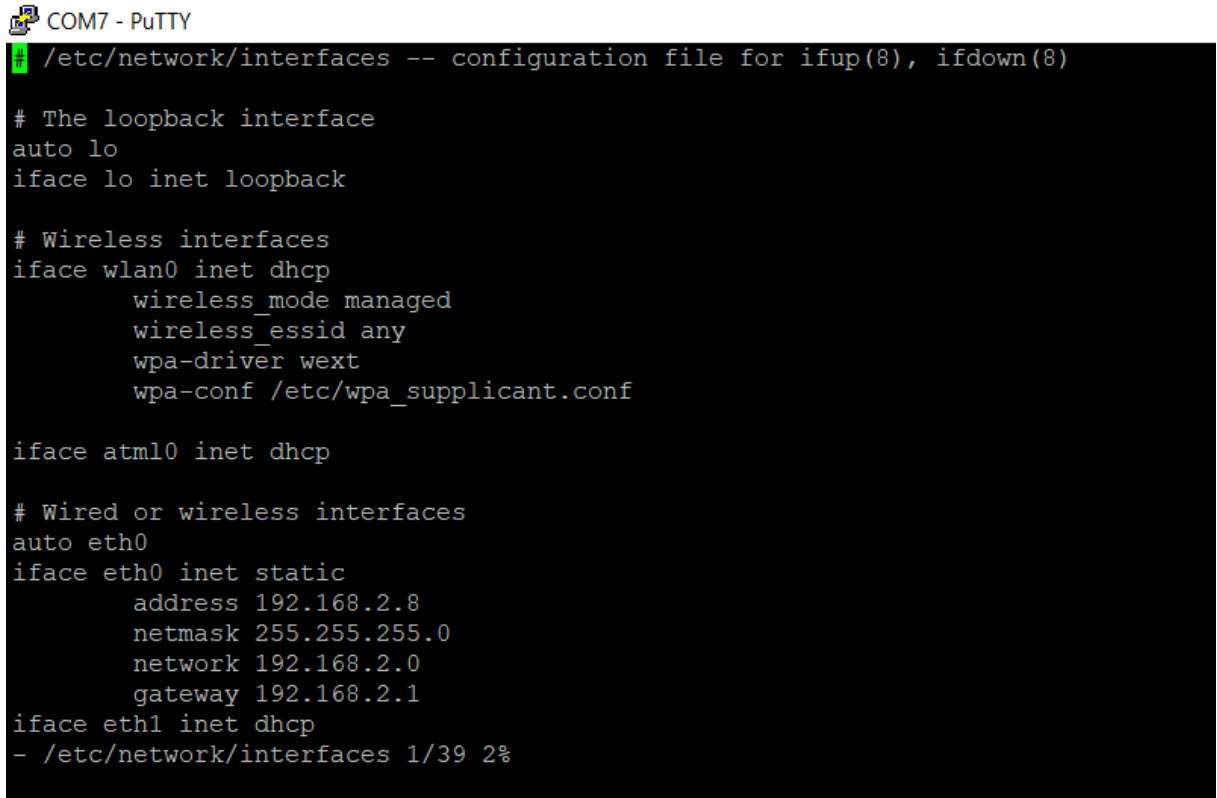
ubuntu@ubuntu:~/Desktop/Embedded_Systems/proje/proje/proje$ petalinux-build
[INFO] Sourcing buildtools
[INFO] Building project
[INFO] Sourcing buildtools extended
[INFO] Sourcing build environment
[INFO] Generating workspace directory
INFO: bitbake petalinux-image-minimal
NOTE: Started PRServer with DBfile: /home/ubuntu/Desktop/Embedded_Systems/proje/proje/build/cache/prserv.sqlite3, IP: 127.0.0.1, PORT: 39627, PID: 228565
WARNING: Host distribution "ubuntu-22.04" has not been validated with this version of the build system; you may possibly experience unexpected failures. It is recommended that you use a tested distribution.
Loading cache: 100% |#####| Time: 0:00:02
Loaded 5128 entries from dependency cache.
Parsing recipes: 100% |#####| Time: 0:00:01
Parsing of 3476 .bb files complete (3471 cached, 5 parsed). 5133 targets, 268 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies
NOTE: Fetching uninative binary shim file:///home/ubuntu/Desktop/Embedded_Systems/proje/proje/components/yocto/downloads/uninative/5ec5a9276046e7eceeac749a18b175667384e1f445cd4526300a41404d985a5b/x86_64-nativesdk-libc.tar.xz;sha256sum=5ec5a9276046e7eceeac749a18b175667384e1f445cd4526300a41404d985a5b (will check PREMIRRORS first)
WARNING: Your host glibc version (2.35) is newer than that in uninative (2.33). Disabling uninative so that sstate is not corrupted.
Initialising tasks: 100% |#####| Time: 0:00:10

```

- Burada bilgisayara bağladığımız Petalinux kurulu sd kartı PYNQ FPGA geliştirme kartına taktıktan sonra gücü verdik. Ardından PuTTY’den seri porttan çıktıları almak için line ve speed ayarlamalarını yapıp seri port bağlantısını açtık.

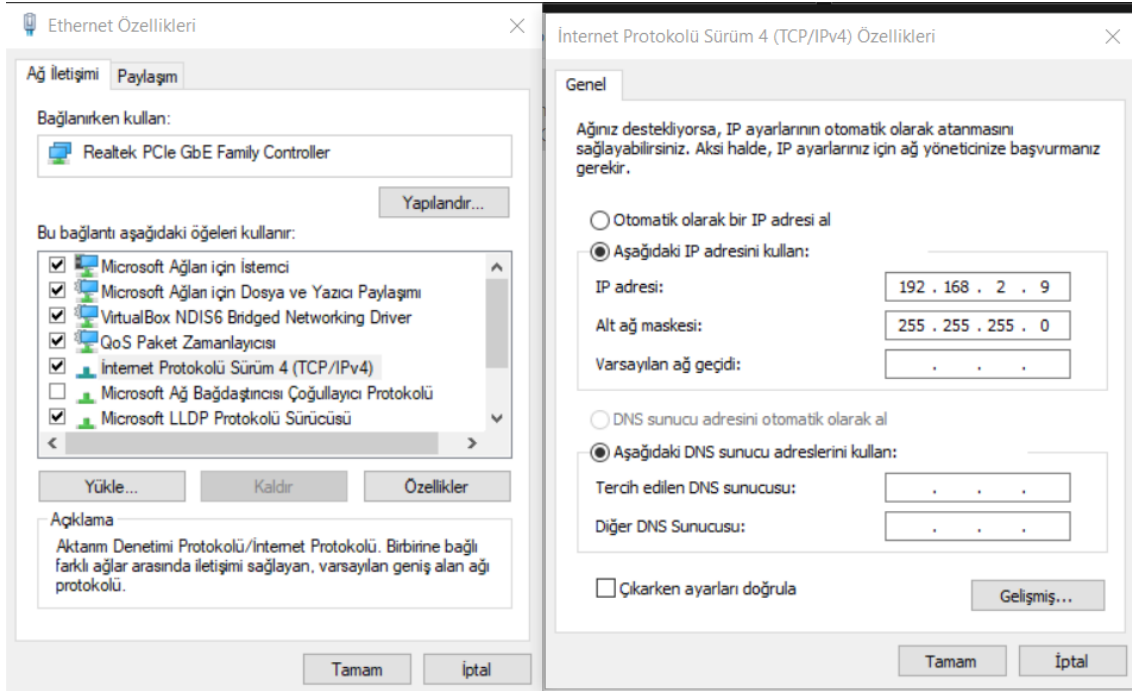


- vi /etc/network/interfaces komutu ile FPGA'in IP atamasını (192.168.2.8) yaptık. Ardından network arayüzünü yeniden başlatmak için 'ifdown -a; ifup -a' komutunu kullandık ve sonrasında da 'ip a' komutu ile FPGA'in IP adresinin gelmiş olduğunu gördük.



```
COM7 - PuTTY
root@projectName:~# mesaj
mesaj
10 20^C
root@projectName:~# ifdown -a; ifup -a
root@projectName:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0a:35:00:1e:53 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.8/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20a:35ff:fe00:1e53/64 scope link
        valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
root@projectName:~#
```

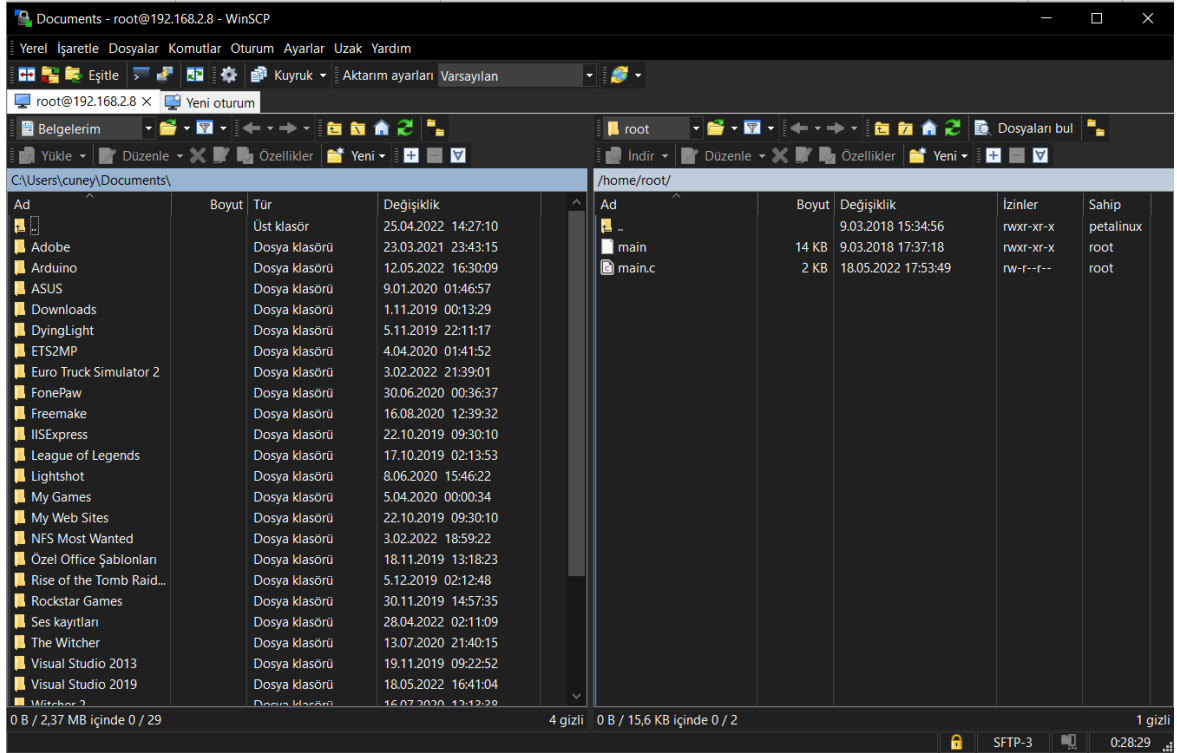
- Ethernet kablosu ile bilgisayar ve PYNQ Kart ile bağlantı kurduktan sonra bağlı olduğumuz Ethernet özelliklerinden, TCP/IPv4'e tıklayarak statik IP(192.168.2.9) verdik.



- Is /dev/tty* komutundan sonra FPGA'imizin seri port arayüzüne sahip olan yer olan ttyPS0'dan veriyi dışarı transfer ettik. "echo" komutu ile ssh arayüzünden basılan mesaj'ların COM4(bilgisayarın seri portu)'ten geldiğini görmüş olduk.

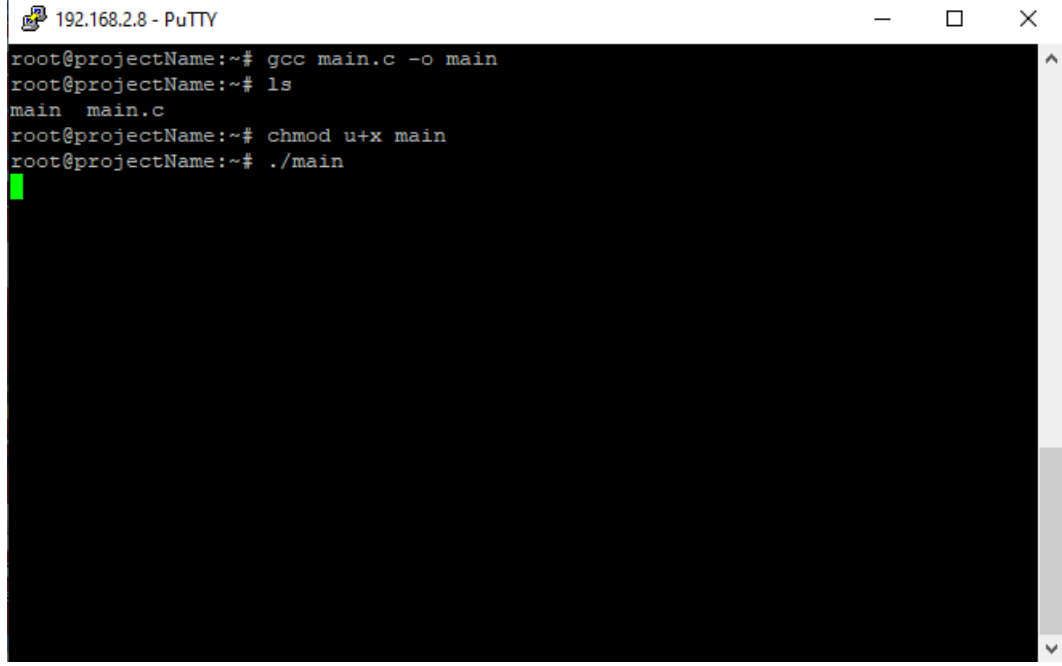
The image shows two PuTTY terminal windows. The left window, titled '192.168.2.8 - PuTTY', shows a root user at a project named 'projectName'. The user enters the command 'echo mesaj > /dev/ttyPS0' twice, and a green cursor is visible on the third line. The right window, titled 'COM7 - PuTTY', shows the same root user at 'projectName' entering the command 'mesaj', and the output 'mesaj' is displayed on the next line.

- Daha sonra WinSCP arayüzünden ethernet altyapısını ayağa kaldıracak kod parçasını ekledik.



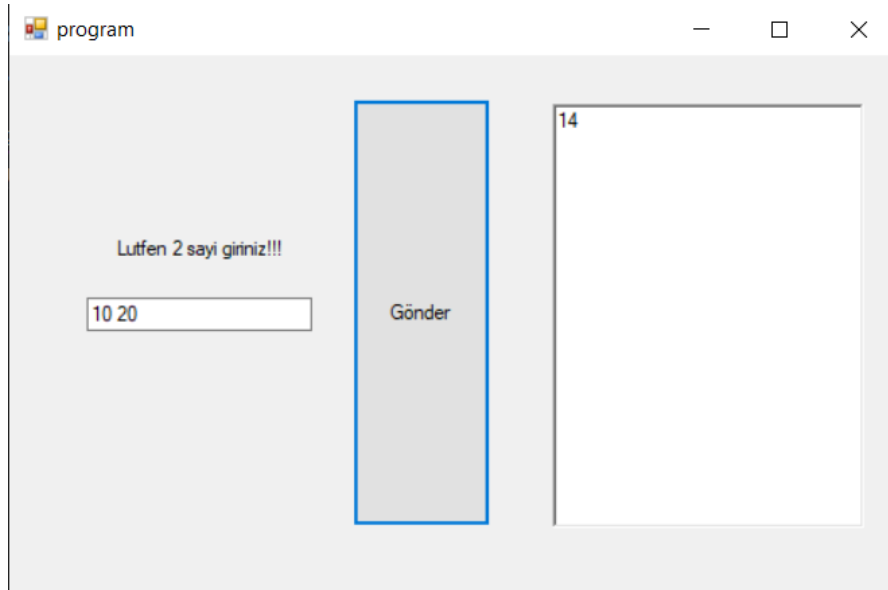
(İçerideki C kodu github'da paylaşılmıştır.)

- Sonrasında ssh'tan 'gcc main.c -o main' komutu ile kodumuzu derledik. Sonrasında 'chmod u+x' komutu ile çalışma yetkisi verdik. Ardından './main' komutu ile çalıştırdık.



```
192.168.2.8 - PuTTY
root@projectName:~# gcc main.c -o main
root@projectName:~# ls
main  main.c
root@projectName:~# chmod u+x main
root@projectName:~# ./main
```

- Daha sonra Visual Studio'da C# ile gerekli olan arayüzü oluşturuyoruz ve petalinux olan FPGA'e göndereceğimiz sayıları girdik.



The screenshot shows a C# application window titled "program". It features a text label "Lutfen 2 sayi giriniz!!!" (Please enter 2 numbers!!!) above a text input field containing "10 20". To the right of the input field is a "Gönder" (Send) button. Further to the right is a large text area displaying the number "14".

IV. Sonular

Petalinux ortamından UART arayüzü üzerinden yakalanan iki sayı girişı verildi ve ardından beklenen ıktının aynı sonucu elde edildi. Test kapsamında sonuç UART arayüzü üzerinden PC'ye gönderildi.

V. Proje Ekibi

Evrin Arda KALAFAT, 25.09.2001 yılında istanbulda doğdu. 2019 yılında Kadıköy Final Temel Lisesi'nden mezun oldu. Şu anda Fenerbahe Üniversitesi Bilgisayar Mühendisliğı bölümünde lisans eğitimi almakta. Bilgisayar Mühendisliğı Bölüm Temsilciğı yapmaktadır. C, C++ ve Pyhton dillerinde bilgili. Programlama, yapay zeka ve siber güvenlik ile ilgileniyor.

Cüneyt BALCI, 28.08.2000 yılında İstanbul'da doğdum. 2018 yılında Final Temel Lisesi'nden mezun oldum. Şu anda Fenerbahe Üniversitesi - Bilgisayar Mühendisliğı bölümünde ve çift ana dal programı kapsamında Ekonomi (İngilizce) bölümünde lisans eğitimi almaktayım. C, C++ ve Python dillerinde bilgili. Network ve blockchain alanlarıyla ilgileniyor.

Arda ALHAN, 18.05.2001 yılında İstanbul'da doğdu. 2019 yılında Eyüp Anadolu Lisesi'nden mezun oldu. Şu anda Fenerbahe Üniversitesi Bilgisayar Mühendisliğı bölümünde lisans eğitimi almakta. C, C++, C#, Selenium ve Pyhton dillerinde bilgili. Programlama ile ilgileniyor.

VI. Referans Dosyalar

VII. <https://youtu.be/VKGzSj22D90>

VIII. https://github.com/rhgod/RISC-V_Project

IX. Kaynaklar

[1] <http://www.levent.tc/courses/embedded-systems>

[2] <http://www.onurbabur.com/linux-sistemlerde-statik-ip-atama-ip-degistirme-islemleri/>