PROJECT #2 GUIDE

UCB ECEN 5803 FALL 2024 PROJECT #2: RASPBERRY PI MODEL 3 QUAD ARM CORTEX A53 WITH WINDOWS 10 IOT AND LINUX

TABLE OF CONTENTS

I. MODULE 1 INSTALL VISUAL STUDIO

1. Follow the directions in VisualStudioInstallationGuide1.pdf to install Visual Studio on your PC. You will need to configure the installation so that it can compile code for your ARM target processor.
2. To prepare for code development for Windows 10 IoT Core, watch the first 20 minutes of each of the following videos:
   https://www.youtube.com/watch?v=isfQEiq-KEo You do not need the display shown, any HDMI monitor will do, including ones in the Lab.
   https://www.youtube.com/watch?v=A-kazyOiBvs Introduction to Windows 10 IoT Core on the Rpi.
   https://www.youtube.com/watch?v=FO-c0FgG4T0 First project, Windows 10 IoT Core on the Rpi.

II. MODULE 2 BOOT WINDOWS 10 IOT, CREATE G.711 CODEC

1. To test Windows 10 IoT, follow the directions here: https://www.makeuseof.com/tag/raspberry-pi-windows-iot-core/ Please be aware that Windows 10 IoT for ARM is still in early stages of development, so some things may not work as you expect. Do not spend more than a couple of hours trying to make it work before reaching for help. A recommended SD Card is SanDisk Ultra Micro SDHC 16GB Also refer to these helpful sites:
   https://docs.microsoft.com/en-us/windows/iot-core/tutorials/rpi
   https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-raspberry-pi-kit-node-get-started
   https://www.raspberrypi.org/documentation/
   https://www.raspberrypi.org/documentation/hardware/raspberrypi/bootmodes/bootflow.md
   https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up
   https://www.raspberrypi.org/documentation/installation/installing-images/windows.md
   https://elinux.org/RPi_Easy_SD_Card_Setup
   https://docs.microsoft.com/en-us/windows/iot-core/manage-your-device/windowsdebugger

https://www.youtube.com/watch?v=_IpjihBU3ps
https://blog.infernored.com/getting-started-with-windows-iot-and-raspberry-pi-3/

2. Upon booting up, the serial port should be sending out debug messages. Open a terminal window to capture them. What do you see? Be aware that the port used to transmit the information may be different from what you expect. You will need to do some research to determine how to connect to it.
3. How much memory is used by the code? (What is the image size?)
4. Capture a screen shot of the terminal window.
5. The Ethernet and HDMI ports should also be active. Connect the HDMI output to a monitor using a HDMI Cable and adapter if necessary, or connect using SSH to see the GUI window. Reboot the system – what do you see?
6. Write C code for a G.711 coder/decoder. See http://www.opensource.apple.com/source/tcl/tcl-20/tcl_ext/snack/snack/generic/g711.c or for an example. Use this decoder to decode a file given to you by your instructor. You will need to use Visual Studio 15 or later to compile code for this application and then run it on the target board. Alternatively, you could also do this in Linux using gcc on the target board.
7. Record your observations. How is the behavior of Windows 10 IoT different from Linux?

III. MODULE 3 SCRIPTING WITH LINUX

Aim: Creation and running an executable script in Linux to print useful information about the Linux Kernel.

Scope: The function implemented by the script is under the discretion of the group. The complexity and number of functions implemented is also under the discretion of the group.

Setup: to create your Raspbian Linux development environment, follow the directions given in Practical Homework 3

Procedure:
Creating an executable script:
i) Use a suitable scripting language for writing a script that uses commands in the terminal for access data relevant to the terminal. Scripts can be written in bash or python or something similar. Information for bash can be obtained from http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html and http://www.freeos.com/guides/lsst/. An introduction to python can be found here: https://docs.python.org/2/tutorial/ and https://www.linuxjournal.com/article/1121 .

ii) Convert the script to an executable file.

iii) An additional 2 bonus points will be awarded if the script is executed at boot time.

Example of what the executable script may perform
- display the current running processes.
- display the current data and time, kernel version
- display the kernel dump.

The points on this exercise will vary on two factors:

A. Number of functions implemented by the script. (Minimum: 3)

B. Complexity of the functions implemented by the script. (A function is not complex if there is 1 exclusive command that performs that function).

IV. MODULE 4 BUILD YOUR OWN PBX WITH ASTERISK

1. For development of code in Linux, GCC and GDB are the preferred compiler and debugger to use. Go to www.asterisk.org and download Asterisk. Look at https://wiki.asterisk.org/wiki/display/AST/Beginning+Asterisk and read the sections on Beginning Asterisk, Installing Asterisk, and the Hello World Project. Use either the Angstrom package repository to install the asterisk package directly after connecting the Raspberry Pi Model 3 to the internet under Linux. Carefully read the documentation and online guides and incorporate Asterisk, the open source PBX into one of your Linux SD cards. How much memory is used by the code? (What is the image size?)
2. Using either a SIP phone plugged into the same LAN as the Raspberry Pi Model 3 (you may need an Ethernet switch to create the network), or with a PC running a softphone application connected to the Raspberry Pi Model 3, configure Asterisk to provide a voicemail message at extension 100. Configure your SIP phone or softphone and register with Asterisk. Show your Asterisk setup in a screenshot.
3. Make a call to extension 100 and record what you hear. Show your Asterisk setup in a screenshot.
4. [Optional, for 5 extra credit points] Add another SIP phone or softphone to the network, and make a phone to phone call.

V. MODULE 5 BUILD A WINDOWS 10 IOT TELECOM OR IOT APPLICATION (OPTIONAL, FOR 10 POINTS EXTRA CREDIT)

1. Using either Windows Embedded Compact 7 or Windows 10 IoT on the Raspberry Pi Model 3, or on a VM on your laptop, create an application related to some telecom or IoT function. You can start with one of the test programs included in the BSP. Some possible applications:

   Any Mobile Phone apps, like SMS messaging, voicemail, or contact manager;

   A Morse Code Receiver that decodes and displays incoming Morse Code generated by a switch on a small microcontroller board;

   An IoT sensor aggregation application that uses SPI, I2C, and/or GPIO sensors and packetizes the information for Ethernet Access;

   A Web server that provides a home webpage for your Raspberry Pi Model 3. At a minimum it should show the time and number of accesses of the page.
2. Build your application, and any associated test hardware or interfaces, and test your new application.
3. Document your new application by providing a zip file of the code and screenshots showing the application in operation. Be prepared to demo this application to the TAs.

VI. MODULE 6 MEASURE CORTEX-A53 DMIPS UNDER WINDOWS 10 IOT AND LINUX (OPT., 5 POINTS EXTRA CREDIT)

1. Port your DMIPs benchmark code from Project 2 and compile and run it under Linux on the Raspberry Pi Model 3. Record you DMIPs numbers using 1, 2, or 4 cores.

2. Port your DMIPs benchmark code from Project 2 and compile and run it under Windows 10 IoT on the Raspberry Pi Model 3. Does it match your expectations? How does it compare to the Linux implementation – do you get the same performance numbers?