Open Embedded

Overview

- Description
- Building from Source
  - Reasons
  - Setting up
  - Procedure

Description

- Linux build system created for the Yocto project
- Very easy to customize as we will show later
- Used primarily for cross-compilation
  - Meaning that you can compile for another operating system
- Works on any Linux distribution

Reasons for Building from Source

- Users knowledgeable in OS can modify the source code
- Console-only images, X11 with GPU accelerated hardware support are some examples of possible customizations
- Another reason is to understand the OS
    - A useful skill to learn for many software applications and opportunities

Host Machine

- Our instructions will revolve around a Linux host machine
- One way to set up the host machine is through a Virtual Machine
  - Make sure that the DragonBoard is connected to your Virtual Machine instead of your actual computer
- Another way is to have your Linux computer
- For all the videos in this lesson, my demo will be on Ubuntu 16.04 on VMWare Workstation Pro

Instructions

- Before starting, familiarize yourself with fastboot
  - Instructions for fastboot is mentioned in a previous video but will be provided again here
- Connect the host computer to DragonBoard
- Set up fastboot mode (steps 1-4)
- 96Boards instruction website will be provided in the readings

Debian

Overview

- Description
- Building from Source
  - Procedure

Description

- Operating system built from free software
- Software licensed under GNU General Public License
- One popular Linux OS based on Debian is Ubuntu
- APT is Debian's way of updating and installing software
  - As mentioned in previous videos, easy to use
- Wide range of support for Debian
  - You will find sudo apt-get install commands for a diverse range of software
- Our choice for most of the Coursera

Building from Source

- Before starting, familiarize yourself with fastboot
  - Instructions for fastboot is mentioned in a previous video but will be provided again here
- Connect the host computer to DragonBoard
- Set up fastboot mode (steps 1-4)
- 96Boards instruction website will be provided in the readings
- Important things to note:
  - Make sure to have space in your host computer (Surprise! Operating systems are big)
  - “<path to your GCC cross compiler>” is the directory’s bin folder
  - If necessary, transfer the module files to /lib/modules folder of the DB410c

Android

Overview

- Description
- Building from Source
  - Procedure

Description

- Operating system intended for mobile phones and tablets
- One of the two most popular OS’s for mobile devices
- Based on a Linux kernel
- Easy app development, which has a large market
- More widely used than Debian or OpenEmbedded
- Interface is intended for touch screen, but still can be used on the DragonBoard

Building from Source

- Before starting, familiarize yourself with fastboot
  - Instructions for fastboot is mentioned in a previous video but will be provided again here
- Connect the host computer to DragonBoard
- Set up fastboot mode (steps 1-4)
- 96Boards instruction website will be provided in the readings
- Important things to note:
  - Make sure to have space in your host computer (Surprise! Operating systems are big)
  - “<path to your GCC cross compiler>” is the directory’s bin folder
  - <KERNEL_TOP> and <ANDROID_TOP> should be substituted with names you can remember

Homework

- Download VMWare Player or Fusion(Mac)
- Read all the instructions for building from source
- Build the images for all three operating systems
- Go on the main websites for Android, OpenEmbedded, and Debian
  - Familiarize yourself with all the operating systems
- Search for BitBake targets

Google Assistant

Overview

- Hardware Requirements
- Description
- Instructions

Hardware Requirements

- DragonBoard with internet access
- Keyboard
- Mouse
- Monitor
- Speakers (eg. HDMI output through monitor)
- Microphone (eg. USB Webcam)

Instructions

- Instructions are provided in the readings
  - Credit to Radhika Paralkar of 96Boards and Google
- Important things to note:
  - When restarting the laptop, type “source venv/bin/activate” again
  - Put the correct path for your .json files
  - Make sure to change audio settings in PulseAudio Volume Control
  - If there is trouble with the libraries, edit the section of the code that imports
    - The fix for me was in lines 31-36

Homework

- Browse through the Google Assistant website
- Browse through the Google Assistant Github
- Read the 96Boards instructions page
- Follow the instructions and run Google Assistant
  - Test Google Assistant with simple requests, such as “Hello” and “Thanks”
- Customize the code in pushtotalk.py

Home Surveillance

Introduction

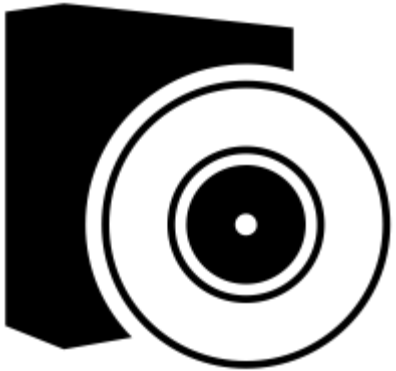
- Created by Manivannan Sadhasivam
- 5 Part in-depth blog about building:
  - https://www.96boards.org/blog/part-1-home-surveillance-project-96boards/
- Can be found in the 96-Boards projects repository:
  - https://github.com/96boards/projects
- Uses AWS S3 for storage and SNS for email notification

Hardware Requirements

- Dragonboard 410c
- Power Supply
- USB Webcam
- Sensors Mezzanine
- Webcam Mount
- Servo X → 5
- Servo Y → 6
    - Note these are different pins from the blog
    - This is so they can be directly connected using Grove Connectors instead of wire
    - Code will need to be changed accordingly

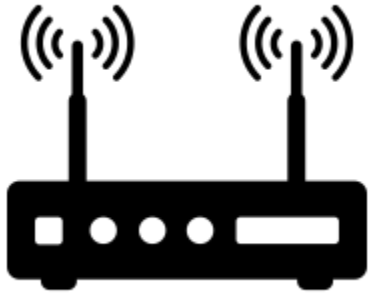
Software Requirements

- OpenCV 3.2.0 (Computer Vision)
- Pillow (Image Processing)
- Arduino
- Flask (Web Server)
- Boto3 (AWS SDK)
- AWS Account



Note

- Port forwarding only required for external access
- We are keeping it on a local network



Description

- Allows you to create and deploy your own personal home surveillance system that will notify via email if there an unwanted person detected.
- Additionally, allows you to access a live stream of your home via your browser

Homework

- Home Surveillance Project Tutorial Parts 1-5
  - Walkthrough the tutorial and run the project
- Home Surveillance Project Github Repo
  - Explore the code
- OpenCV - Facial Detection
  - Read up on how facial detection works
- OpenCV - Facial Recognition
  - Read up on some of the facial recognition classes
- AWS - S3 Python API
  - Read up on how to use the API