Hands-on Workshop: Introduction to Embedded Linux Applications

Larisa Radu, Dariana Roman
Software Engineer, Development Engineering

Monica Ignat
Applications Engineering Manager

analog.com

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Who are we?

Founded	**1965**
Headquarters	**Norwood, MA**
Employees	**~26,000**
Countries	**30+**
Products	**~75,000 SKUs**
Customers	**125,000**
Publicly Listed	**NASDAQ:ADI** Part of S&P 500 and NASDAQ 100
Design Centers	**~45**
Global Manufacturing	**U.S. (Massachusetts, California, Washington), Ireland, Philippines, and Malaysia**

Over 8000 patents and $4 billion R&D investment in the past 10 years.

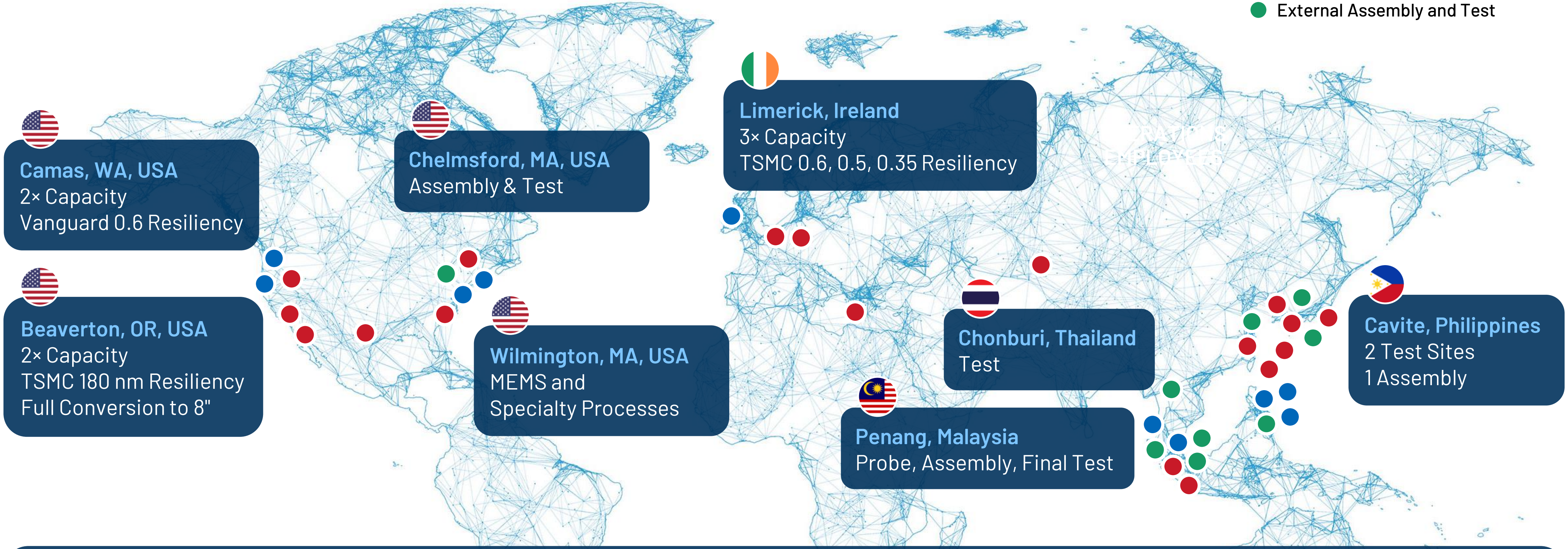
ADI’s Global Footprint

Improves Resiliency and Flexibility

● ADI Internal Sites

● Foundries

● External Assembly and Test



Camas, WA, USA
2× Capacity
Vanguard 0.6 Resiliency

Beaverton, OR, USA
2× Capacity
TSMC 180 nm Resiliency
Full Conversion to 8"

Chelmsford, MA, USA
Assembly & Test

Wilmington, MA, USA
MEMS and
Specialty Processes

Limerick, Ireland
3× Capacity
TSMC 0.6, 0.5, 0.35 Resiliency

Chonburi, Thailand
Test

Penang, Malaysia
Probe, Assembly, Final Test

Cavite, Philippines
2 Test Sites
1 Assembly

- Doubling capacity at internal fabs in U.S. and Europe
- Expanding partnership with TSMC
- Assembly and Test resilient to Taiwan/China

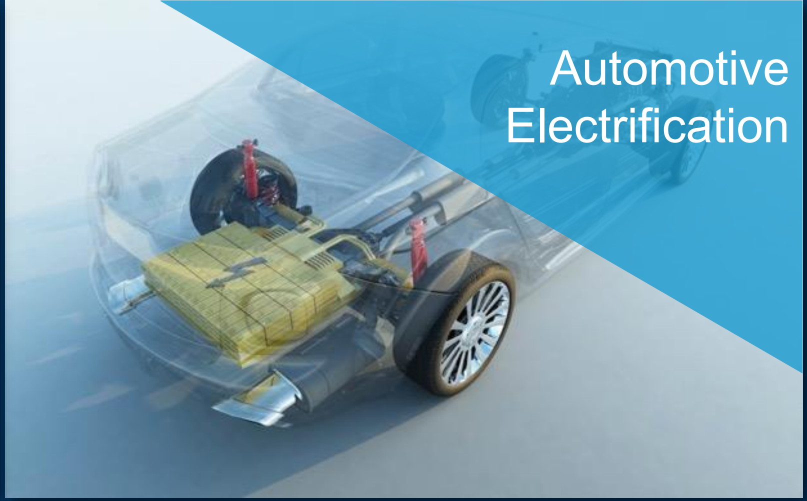
15k+
OPERATIONS EMPLOYEES

11
INTERNAL FACTORIES

50+
SUPPLY CHAIN FACTORIES ACROSS 8 COUNTRIES

Technologies for Today's and Tomorrow's Innovations

Autonomous Transportation and Machines



Automotive Electrification

5G and Next-Gen Connectivity

Digital Health

Industry 4.0 and Smart Energy

Immersive Consumer Experiences

ADI Romania Design Center

Founded in 2011

Office 1 - UBC Riviera

- 1,000 square meters, 100 people capacity

Office 2 - UBC Tower

- 1,000 square meters, 120 people capacity

Multidisciplinary team

- Hardware design
- FPGA development (VHDL, Verilog)
- Embedded software (C/C++, Linux)
- Applications software (Python, MATLAB, C++)
- Devops (Jenkins, Microsoft Azure, CI/CD)
- System architecture
- UX design
- Program/Project management

Project fields

- RF Communications
- Precision & High-Speed Instrumentation
- Depth, Perception and Ranging (ToF, LIDAR)
- Industrial Automation

Hands-on Workshop: Introduction to Embedded Linux Applications

Agenda

Part 1

- Theoretical Background – what is Linux
- Why Linux on embedded systems
- Linux structure
  - User space
  - Kernel space
- Device tree
- What is a Linux driver

Part 2

- Hands-on activity
- Q&A session

Theoretical background – What is Linux?

Linux OS – what is it

- Meet Tux
- Operating system found everywhere

Why Linux

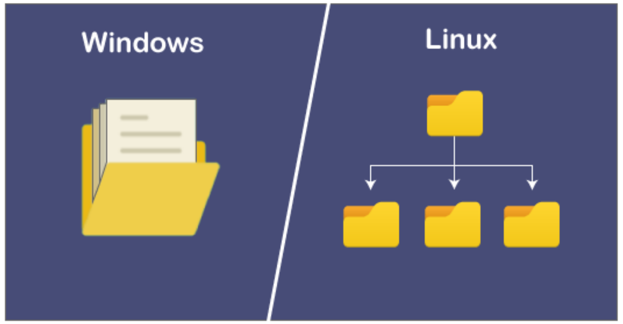
- Free
- Open source
- No safety issues

Components

- Bootloader
- Kernel
- Init system
- Daemons
- Graphical server
- Desktop environment
- Applications

Theoretical background – Linux vs Windows

Linux	Windows
Security	Prone to malware
Open source	Closed source
Community developed	Developed by a company
Free	License based
Lightweight	Resource intensive



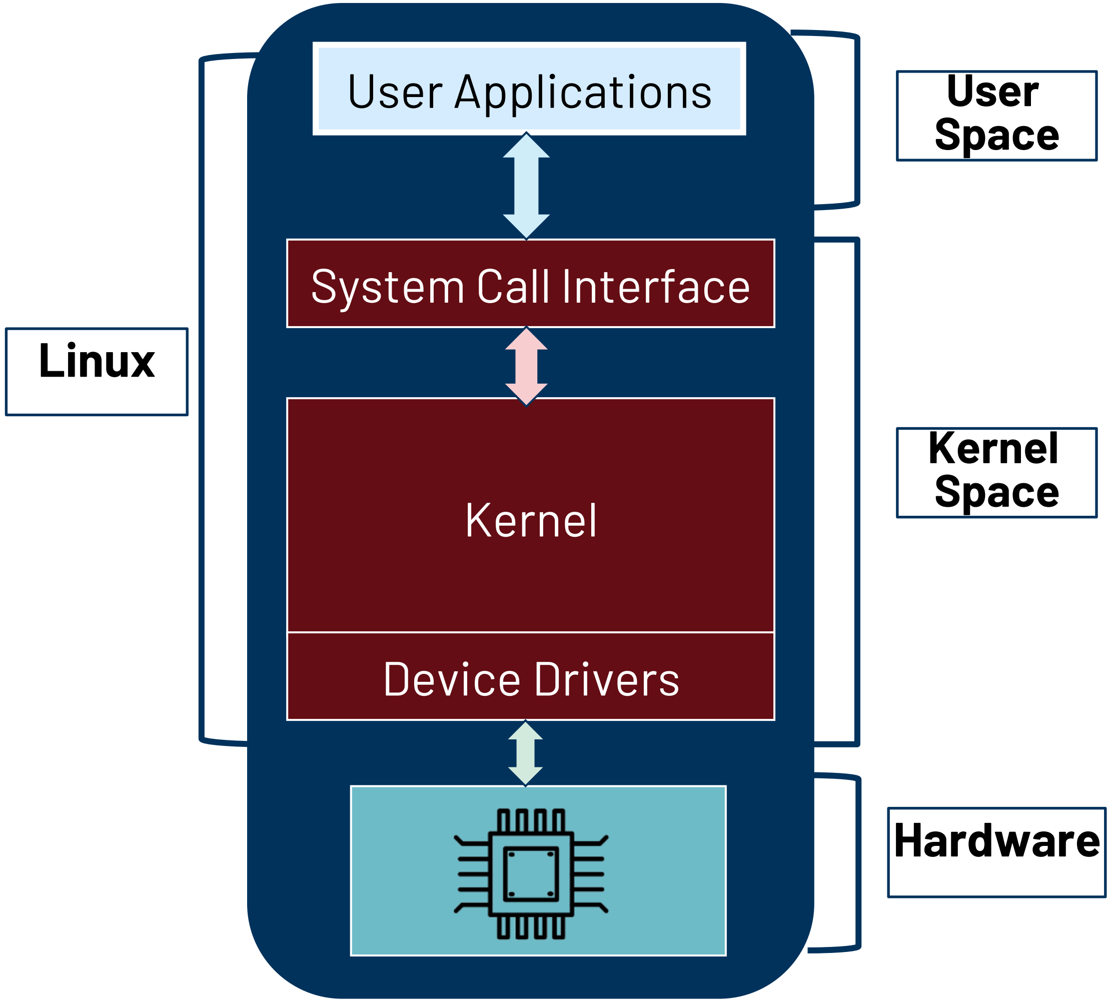
Theoretical background – why Linux on embedded systems?

- Linux can be used in reduced size systems
- Lightweight storage requirements
- Robust
- Stabil
- Large developers' community
- Support for a wide range of software architectures
- Ability to run on fewer resources
- Vendor independence

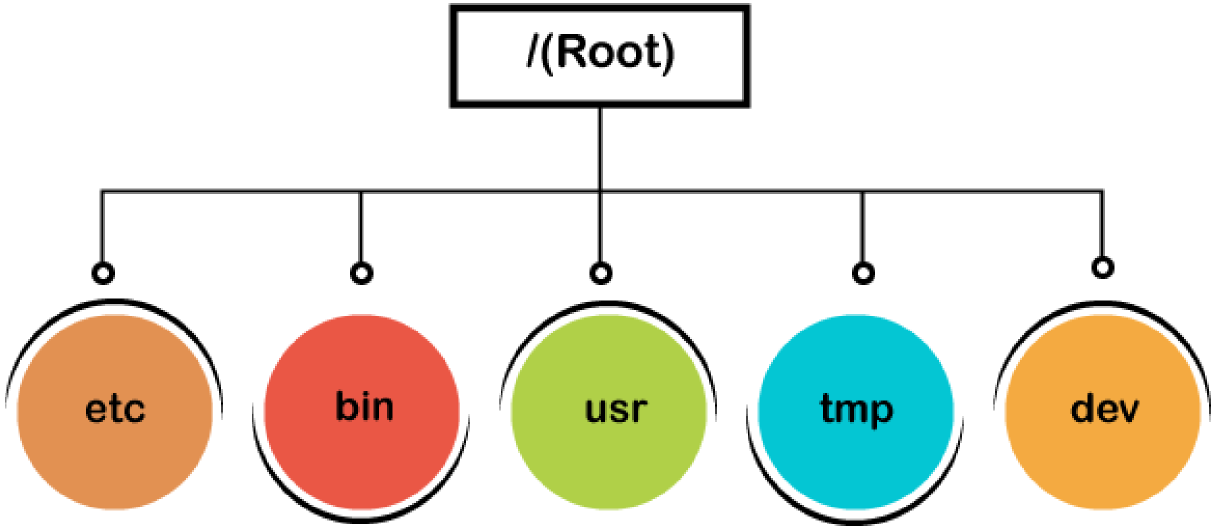
Theoretical Background – Linux Structure

User space – the area of memory that non-kernel applications run in, executed by a user in the operating system, rather than being part of the operating system itself.

Kernel space – the area of system memory reserved for the kernel



Theoretical background – User Space



Distros	Debian based	Ubuntu
		Rpi
		Kuiper
	Arch	Manjaro
	RPM	Fedora
		CentOS

The root filesystem is at the top of the hierarchical file tree (also known as '/'). It has a number of directories containing files critical for booting the system and system operations.

It contains C libraries and user applications.

User space processes run in the user space part of the memory.

The user mode is the non-privileged execution mode that the instructions are executed with.

https://www.javatpoint.com/linux-vs-windows

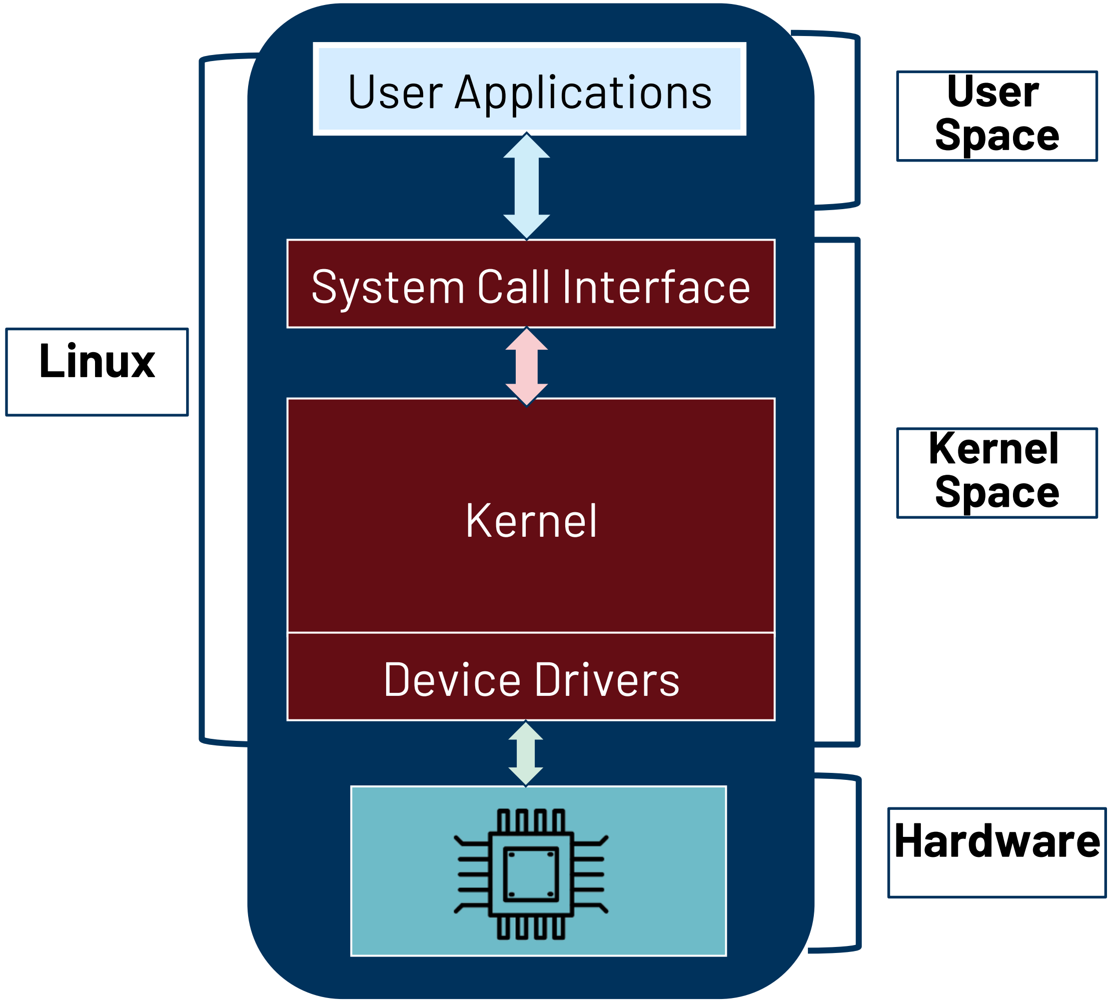
Theoretical background – Linux Kernel (Space)

Distros agnostic - various distros are built on the same kernel

The kernel is loaded when OS is loaded and remains in memory until the OS is shut down.

Contains the System Call Interface, the Linux Kernel and the device drivers.

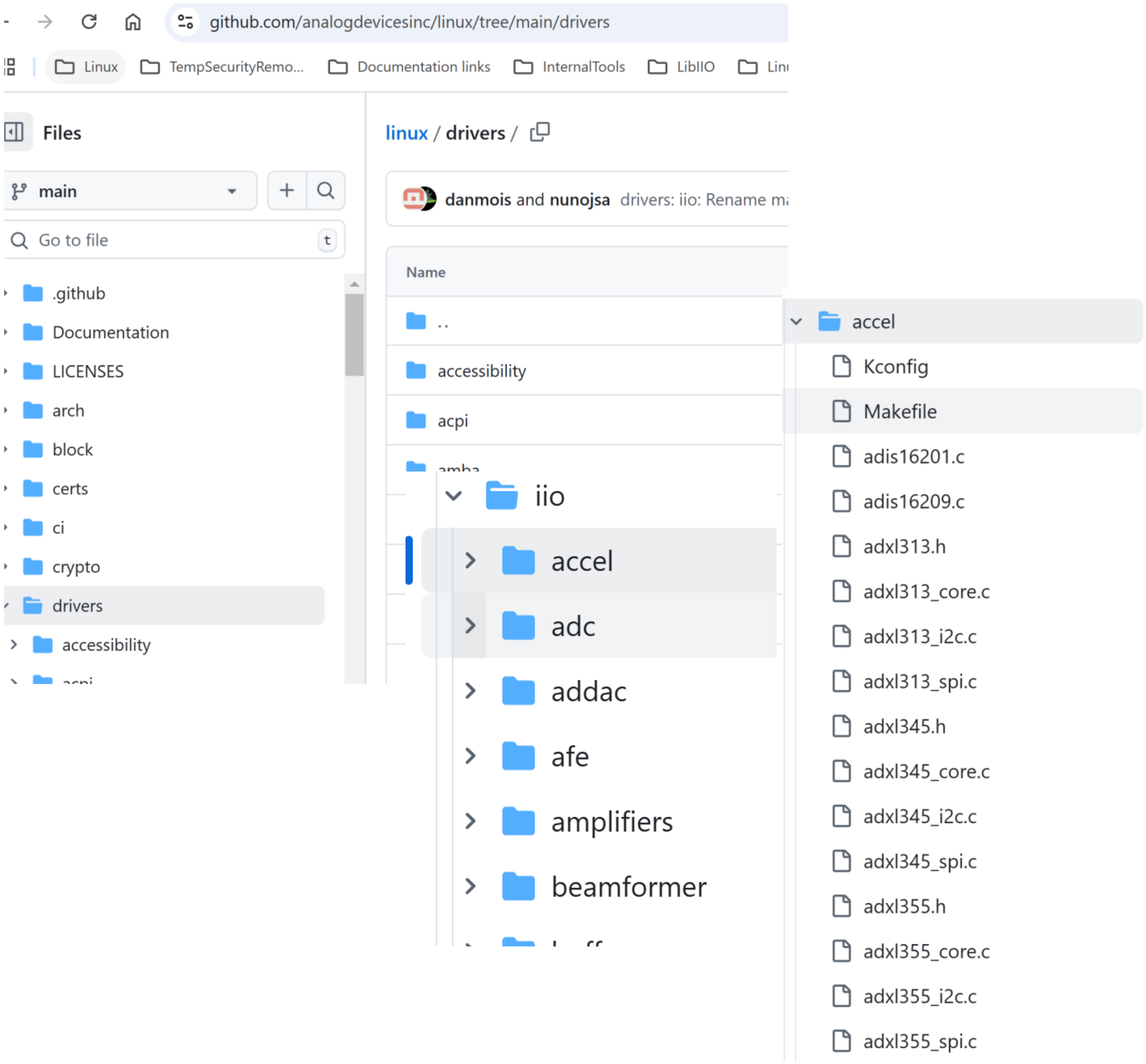
Linux drivers need to be included in the kernel so the hardware becomes accessible in the user space



Linux Device Drivers

What is a device driver?

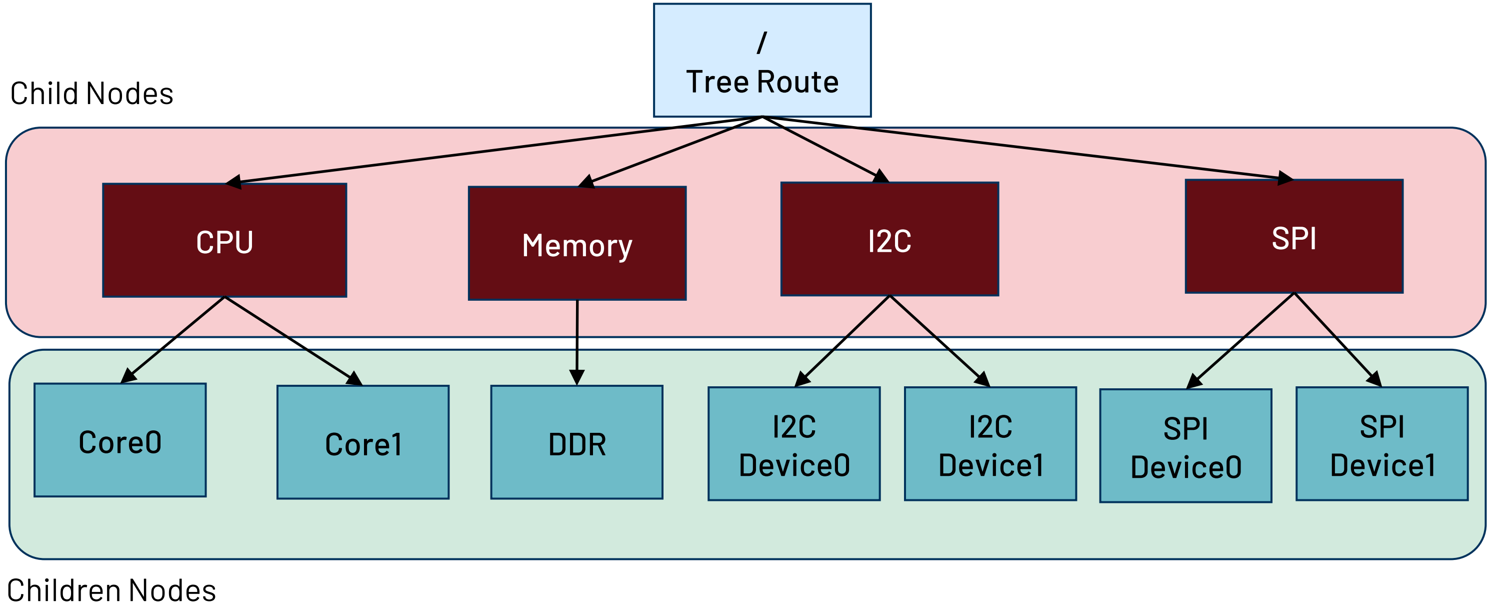
- a code which knows how to handle a particular device. It gives the right interfaces for the user-space to access that device and the device driver code also abstracts various hardware details from the user-space applications
- Implemented in C, in a .c and .h pair, stored under /drivers/"type"/
- It's compiled with kernel



Theoretical background – Device Tree (hardware abstraction)

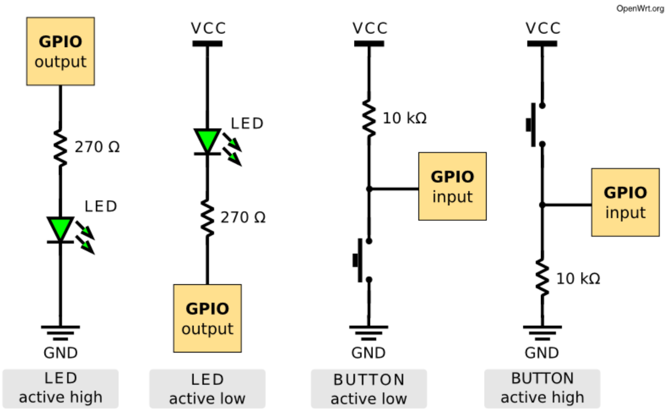
Devicetree (DT) is a data structure and language for describing hardware. Structurally, the DT is a tree, or acyclic graph with named nodes. Its purpose is to describe and configure the hardware connected to a system.

Block Diagram of a simple device tree structure



Theoretical background – GPIO Interface

GPIO stands for **General-Purpose Input/Output** and refers to a set of pins on a device(IC). These pins can send or receive electrical signals, but they aren't designed for any specific purpose. This is why they're called "general-purpose" IO.



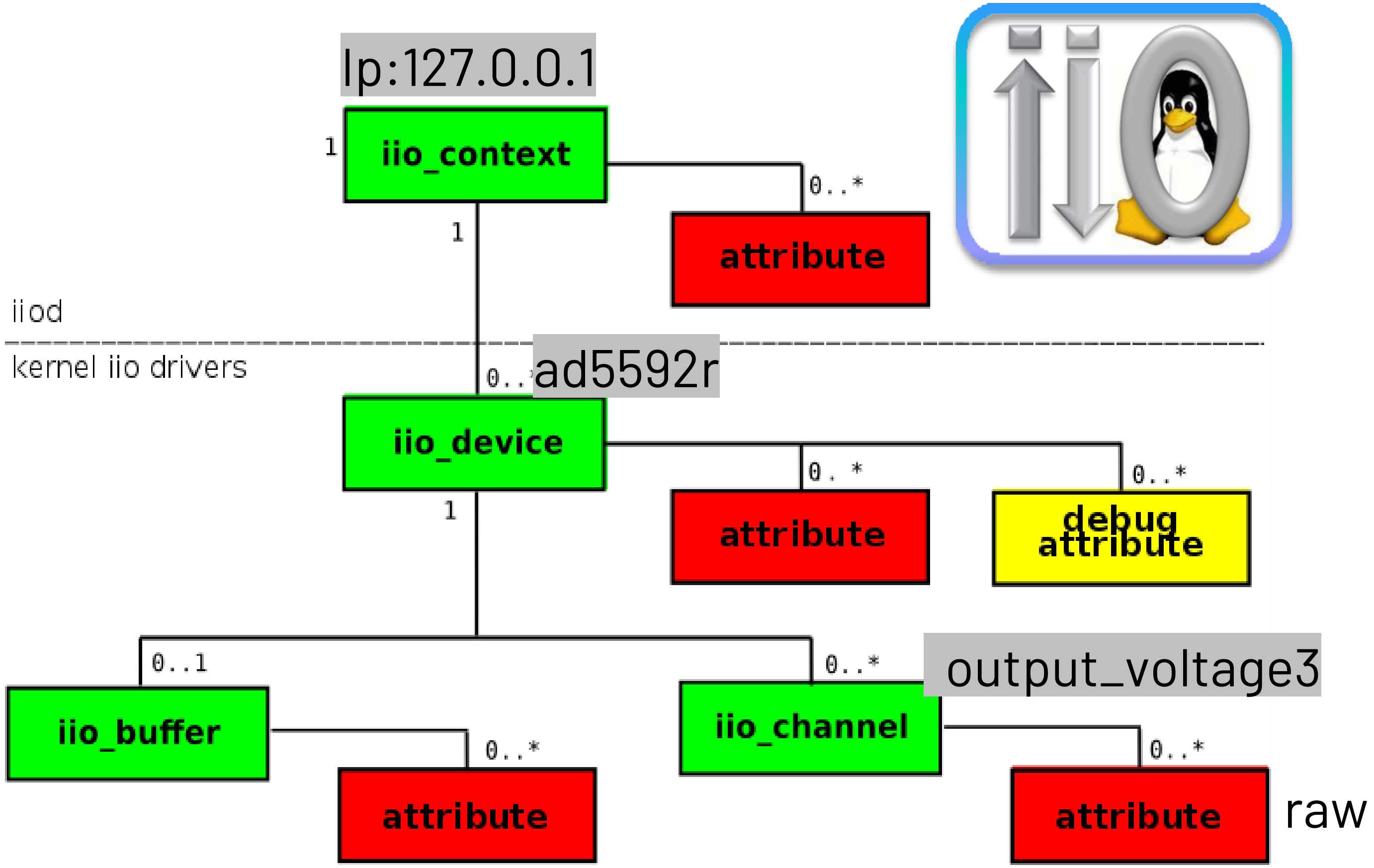
IIO Concepts

Linux kernel **I**ndustrial **I**nput / **O**utput framework

- Not really just for Industrial IO
- All non-HID IO
- ADC, DAC, TRX, light, accelerometer, gyro, magnetometer, humidity, temperature, pressure, rotation, angular momentum, chemical, health, proximity, counters, amplifiers, synthesizers, etc.

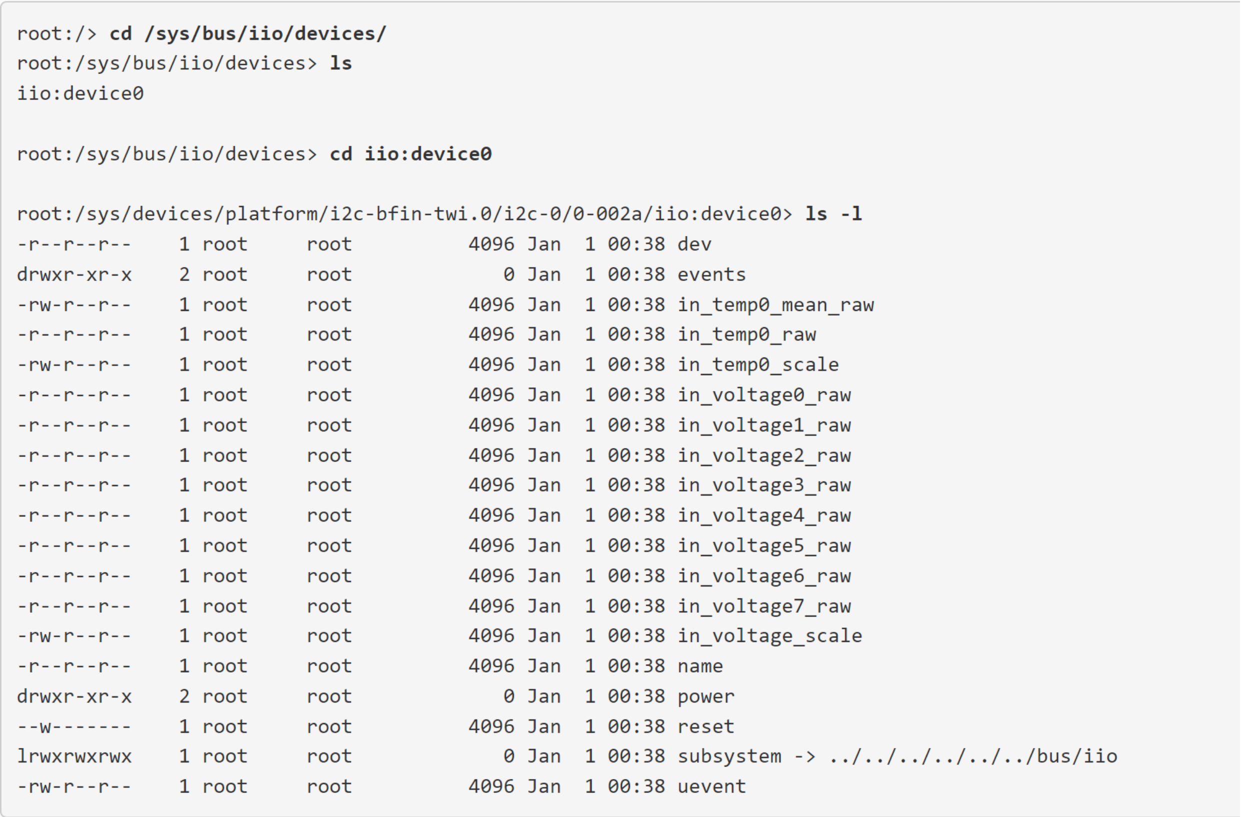
In the upstream Linux kernel for more than 10 years.

****CROSS Platform Libiio library, with LOTS of language bindings ****



IIO Concepts

- The Linux Industrial I/O (IIO) subsystem is intended to provide support for devices that, in some sense, are analog-to-digital or digital-to-analog converters
  - Devices that fall into this category are:
    - ADCs
    - DACs
    - Accelerometers, gyros, IMUs
    - Capacitance-to-Digital converters (CDCs)
    - Pressure, temperature, and light sensors, etc.
    - RF Transceivers (like the AD9361 / AD9364 / AD9371 / ADRV9009)
  - Can be used on ADCs ranging from a 1MSPS SoC ADC to >5 GSPS ADCs

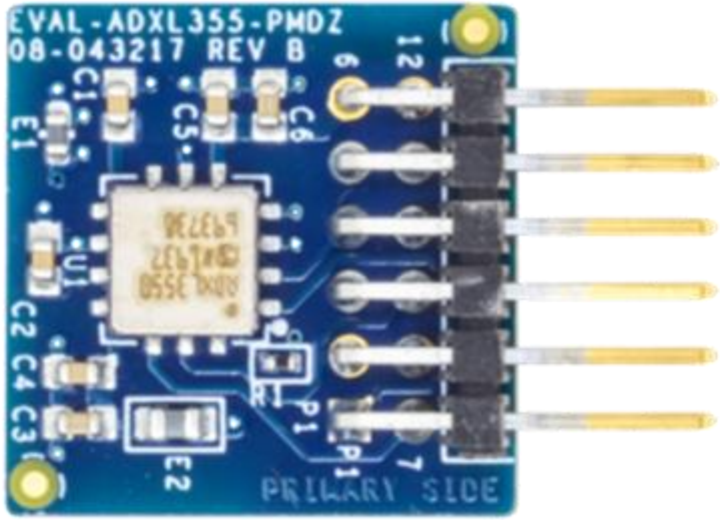


```
root:/> cd /sys/bus/iio/devices/
root:/sys/bus/iio/devices> ls
iio:device0

root:/sys/bus/iio/devices> cd iio:device0

root:/sys/devices/platform/i2c-bfin-twi.0/i2c-0/0-002a/iio:device0> ls -l
-r--r--r--    1 root     root          4096 Jan  1 00:38 dev
drwxr-xr-x    2 root     root             0 Jan  1 00:38 events
-rw-r--r--    1 root     root          4096 Jan  1 00:38 in_temp0_mean_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_temp0_raw
-rw-r--r--    1 root     root          4096 Jan  1 00:38 in_temp0_scale
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage0_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage1_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage2_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage3_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage4_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage5_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage6_raw
-r--r--r--    1 root     root          4096 Jan  1 00:38 in_voltage7_raw
-rw-r--r--    1 root     root          4096 Jan  1 00:38 in_voltage_scale
-r--r--r--    1 root     root          4096 Jan  1 00:38 name
drwxr-xr-x    2 root     root             0 Jan  1 00:38 power
--w-------    1 root     root          4096 Jan  1 00:38 reset
lrwxrwxrwx    1 root     root             0 Jan  1 00:38 subsystem -> ../../../../../../bus/iio
-rw-r--r--    1 root     root          4096 Jan  1 00:38 uevent
```

ADXL355 part

Low Noise, Low Drift, Low Power, 3-Axis MEMS Accelerometers

- ADXL355 digital output features
  - Digital SPI and I^{2}C interfaces supported
  - 20-bit ADC
  - Data interpolation routine for synchronous sampling
  - Programmable high- and low-pass digital filters
- 0 *g* offset vs. temperature (all axes): 0.15 m*g*/°C maximum
- V_{SUPPLY} with internal regulators: 2.25 V to 3.6



Hands-on Workshop: Introduction to Linux Embedded Applications

Part 2
Hands on Activity

Hands On Activity

By the end of this lab, you will learn:

- How to **interact with Linux filesystem**
- How to use a Linux demo running on a specific hardware
- How to modify written code to do a certain task

http://www.iconarchive.com/show/noto-emoji-objects-icons-by-google/62807-radio-icon.html
http://www.streamlineicons.com
http://pixelkit.com

Hands On Activity – Getting started steps

Insert the provided USB stick into an USB port of your computer.

Turn off your computer, then turn it on and at the same time keep pressing on the F12 key 2-3 times a second to enter BIOS.

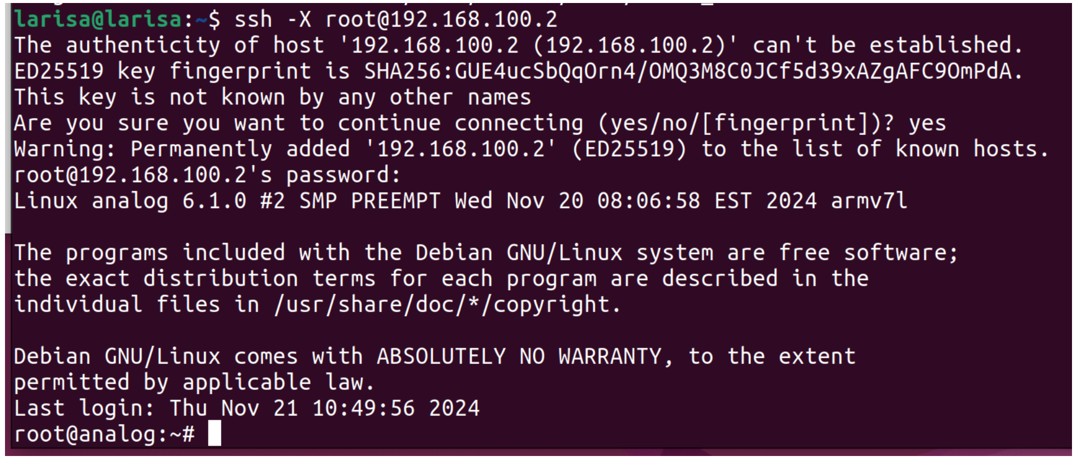
Select the USB option when asked what device to boot from.

Select the first option (Debian GNU/Linux Live) and wait for Linux to boot.

Click *Activities* (upper-left corner), search for *Terminal* and open it.

Run the following command in the terminal: *ssh -X root@192.168.100.2*

At this point you should see the following output in the serial terminal:



```
larisa@larisa:~$ ssh -X root@192.168.100.2
The authenticity of host '192.168.100.2 (192.168.100.2)' can't be established.
ED25519 key fingerprint is SHA256:GUE4ucSbQqOrn4/OMQ3M8C0JCf5d39xAZgAFC9OmPdA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.2' (ED25519) to the list of known hosts.
root@192.168.100.2's password:
Linux analog 6.1.0 #2 SMP PREEMPT Wed Nov 20 08:06:58 EST 2024 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 21 10:49:56 2024
root@analog:~#
```

Hands On Activity

Equipment used:

- EVAL-ADXL355-PMDZ
- Cora Z7 FPGA Platform
- USB cable
- Ethernet cable

Exercise 1

Turn on a LED and make it blink with a heartbeat rhythm

Exercise 2

Run a script

Exercise 3

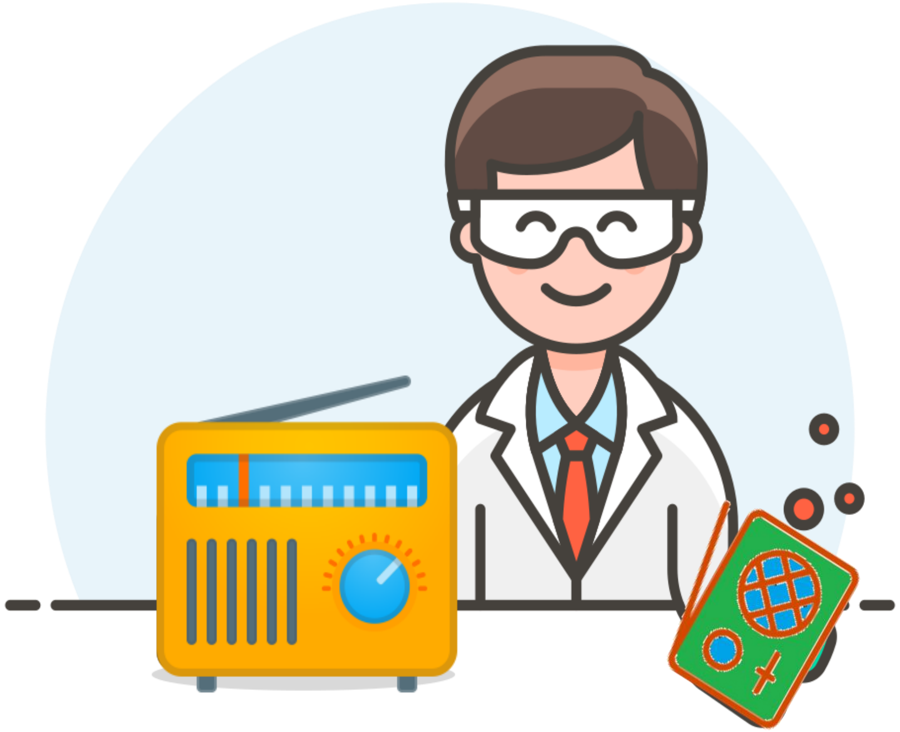
Build a kernel object for ADXL355 driver

Exercise 4

Compute the temperature value from sysfs

Exercise 5

Play a Snake game with the accelerometer



http://www.iconarchive.com/show/noto-emoji-objects-icons-by-google/62807-radio-icon.html
http://www.streamlineicons.com
http://pixelkit.com

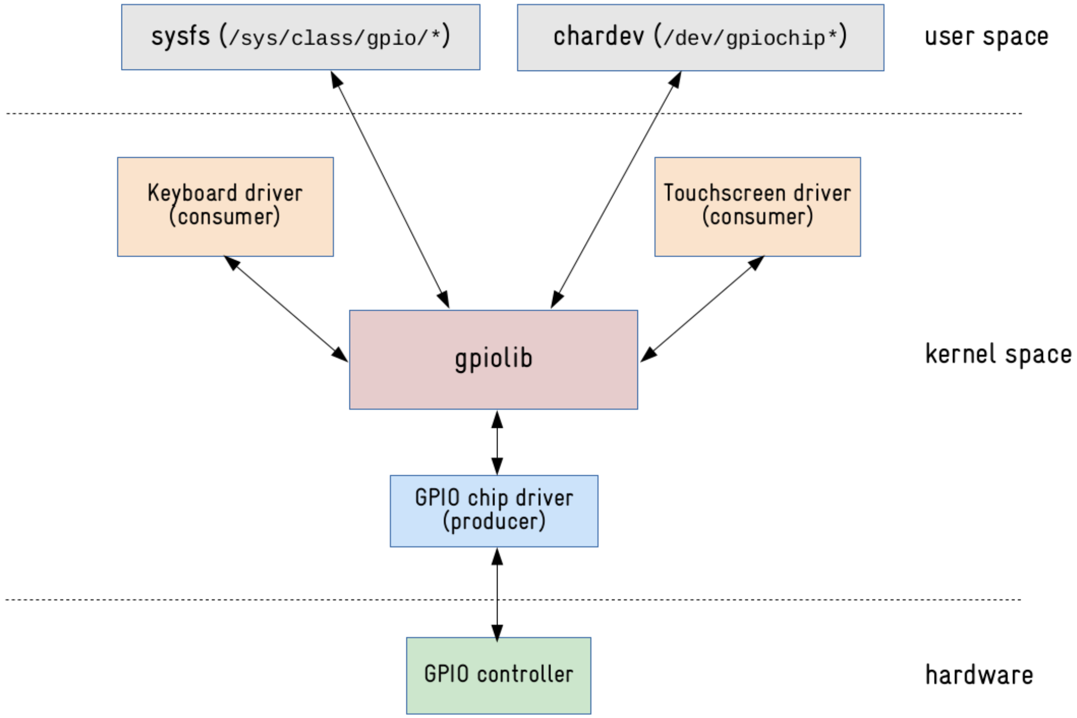
Hands On Activity -Theoretical background – GPIO Interface

GPIO stands for **General-Purpose Input/Output** and is one of the most commonly used peripherals in an embedded Linux system. The GPIO access is done using the GPIOLIB framework

Sysfs is a pseudo filesystem provided by the Linux kernel that makes information about various kernel subsystems, hardware devices, and device drivers available in user space through virtual files. GPIO devices appear as part of sysfs.

The basic steps to use a GPIO pin from the sysfs interface:

- Export the pin.
- Set the pin direction (input or output).
- If an output pin, set the level to low or high.
- If an input pin, read the pin's level (low or high).
- When done, unexport the pin.



sysfs (/sys/class/gpio/*)
chardev (/dev/gpiochip*)
user space
Keyboard driver (consumer)
Touchscreen driver (consumer)
gpiolib
kernel space
GPIO chip driver (producer)
GPIO controller
hardware

Hands On Activity

Exercise 1

Turn on a LED and make it blink with a heartbeat rhythm

Setup

a. Connect the AXDL355 to the JA port of the CORA Z7 board

b. Connect the Cora Z7 UART port of the USB port of your computer using the USB uC cable received

c. Connect the Cora Z7 Ethernet port to the PC Ethernet port.

Procedure:

- **Move to the LEDs directory suing the below command:**
  - *$ cd /sys/class/leds*
- **Turn on the red color of the first LED**
  - $
- **Move to the red directory of the first LED:**
- $ cd led1_red
- **Set the *brightness* to be 1:**
- echo 1 >> led0_red/brightness
- **Make the red color of the first led blink in heartbeat mode**
- echo heartbeat >> led0_red/trigger

Hands On Activity

Exercise 1

Turn on a LED and make it blink with a heartbeat rhythm

Challenge

Make the first LED light continuously purple and the second LED light green for 1000 ms OFF and for 500 ms ON.

Challenge Solution

- **led0:**
  # echo none >> led0_red/trigger
- # echo 1 >> led0_red/brightness
- # echo 1 >> led0_blue/brightness

- **led1:**
  echo 1 >> led1_green/brightness
- echo timer >> led1_green/trigger
- echo 1000 >> led1_green/delay_off
- echo 500 >> led1_green/delay_on

Hands On Activity

Exercise 2

Run a script

Procedure:

- Run the script:
- bash /leds.sh
- Make the script available to be run from anywhere in the file system and run it again
- # chmod +x /leds.sh
- # mv /leds.sh /usr/local/bin
- # leds.sh

Hands On Activity

Exercise 3

Build a kernel object for ADXL355 driver

Procedure:

- Move to the *adxl355* folder:
- # cd /adxl355
- Run *make* to build the kernel object:
- # make
- Load the new kernel object:
- # insmod adxl355_spi.ko
- Check again with "*iio_info*" to see readings from the accelerometer

Hands On Activity

Exercise 4

Compute the temperature value from sysfs

The formula for the temperature:

$$TEMPERATURE = (RAW + OFFSET) \cdot SCALE$$

Procedure:

- **Move to the iio:device folder:**
  - # cd /sys/bus/iio/devices/iio:device1

Here you can see all the attributes of the adxl355:

  - # ls

1. **Put temperature values in variables**:
   - # SCALE=$(cat in_temp_scale)
   - # OFFSET=$(cat in_temp_offset)
   - # RAW=$(cat in_temp_raw)
2. **Compute the temperature and display it in the terminal (keep in mind that it will be in millidegrees Celsius):**
   - # echo "($RAW + $OFFSET) * $SCALE" | bc

Hands On Activity

Exercise 5

Play a Snake game with the accelerometer

Procedure:

- **Run the game controls in the background:**
- # python3 /game.py &
- **Start the Snake game:**
- # /usr/games/snake
- **Move the accelerometer board around and observe the output. Be careful not to disconnect the wires**.

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Resources

Linux:

https://wiki.analog.com/resources/tools-software/linux-drivers-all

https://github.com/analogdevicesinc/linux/tree/main

kernel.org

Specific hardware resources:

https://wiki.analog.com/resources/eval/user-guides/eval-adxl355-pmdz/no-os-setup?s[ ]=no&s[ ]=os#adxl355_driver

AHEAD OF WHAT'S POSSIBLE™

Opportunities at ADI

- Internships
- Jobs
- Summer practice
- Get hardware and support from ADI to develop your own projects

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Our departments:

- Hardware Design
- FPGA Digital Design
- Embedded Software
- Applications Software
- Applications Engineering

Analog Devices Summer Practice Registration

Jobs

Send us your CV!

To: *office.romania@analog.com*
Subject: Internship/Practica

Thank You!
Questions?