João Gameiro, 93097
Pedro Abreu, 93240

Arquiteturas para Sistemas Embutidos

Aula6

NVIDIA Jetson Nano Developer Kit is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. Jetson is a low-power system and is designed for accelerating machine learning applications. [1]

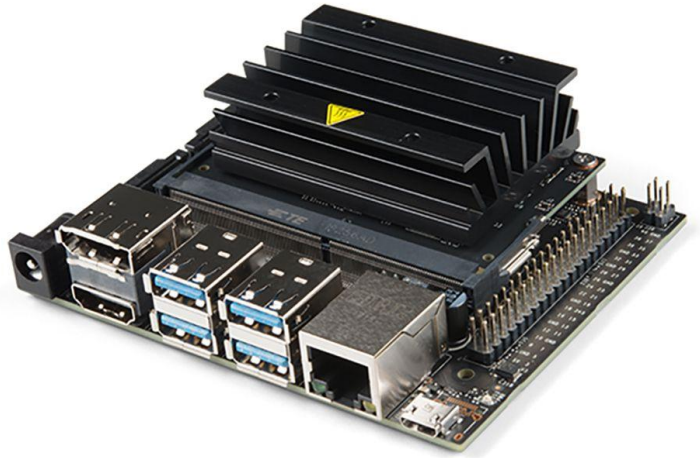

Figure 1: NVIDIA Jetson Nano

ARM Cortex-A57 Processor

[2]

The Cortex-A57 processor is a high-performance, low-power processor that implements the ARMv8 architecture. It has one to four processors in a single multiprocessor device, or MPCore device, with L1 and L2 cache subsystem and a superscalar, variable-length, out-of-order pipeline. It supports static and dynamic branch prediction with a Branch Target Buffer (BTB) and a Global History Buffer (GHB) RAMs, a return stack, and an indirect predictor.

It supports the A32, T32 and A64 instruction sets as well as the AArch32 and AArch64 execution states. Contains functional units for instruction Fetch, Instruction Decode, Instruction Dispatch, Execution of Integer Operations and advanced SIMD and Floating-Point unit (illustrated in Figure2).

In terms of memory systems three are identifiable (also represented in Figure 2):

- Load/Store Units
  - Data cache of 32KB, 2-way set-associative
  - 32-entry fully-associative L1 data TLB with native support for 4KB, 64KB, and 1MB page sizes
- L2 Memory System
  - Services the L1 instruction and data caches misses
  - L2 cache 512KB, 1MB or 2MB, 16-way set-associative cache
  - Duplicate copy of L1 data cache Tag RAMs from each processor
  - Automatic hardware prefetcher
- L1 Memory System
  - Instruction cache 48KB 3-way set-associative
  - Data cache 32KB 2-way set-associative
  - Both with LRU replacement policy

The ARM Cortex-A57 also contains a set of additional components which include:

- **Generic Timer** that provides the ability to schedule events and trigger interrupts.
- **Generic Interrupt Controller (GIC)** CPU interface for supporting and managing interrupts.
- **Advanced SIMD and Floating-point unit** provides support for the ARMv8 Advanced SIMD and Floating-point execution
- The **ETM (Embedded Trace Macrocell)** is a module that performs real-time instruction flow tracing based on the ARM specifications
- **PMU Performance Monitor Unit** to gather statistics about the operation of the processor and memory system during runtime

There is also the possibility for an additional cryptography engine that's not included in the A75 Processor that would be used to perform cryptographic operations.

However FP and Advanced SIMD units provide support for the cryptographic operations of the engine.

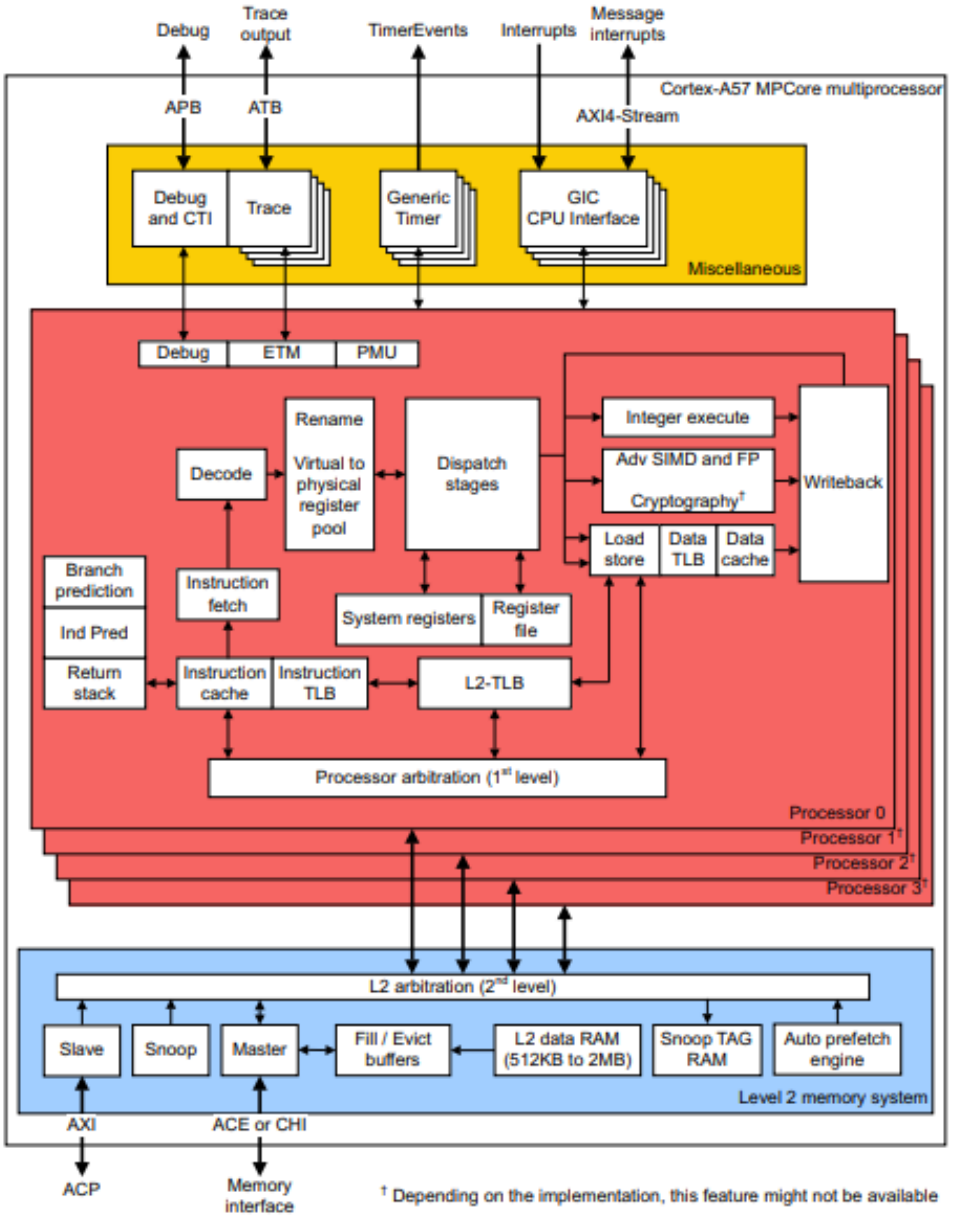

Figure 2: Cortex A57 MPCore multiprocessor

GPU - NVIDIA Maxwell

The NVIDIA Maxwell GPU is a dedicated hardware block for rasterization, shading, texturing, and computation of graphics processing. It's NVIDIA's next-generation architecture for CUDA compute applications and introduces an all-new design for the Streaming Multiprocessor (SM) that dramatically improves energy efficiency. Improvements to control logic partitioning, workload balancing, clock-gating granularity, compiler-based scheduling, number of instructions issued per clock cycle, and many other

enhancements.

The new Streaming Maxwell Multiprocessor (SMM) contains a quadrant-based design with four 32-core processing blocks (which means each SMM contains 128 cores). Each block contains a dedicated warp scheduler capable of dispatching two instructions per clock and its execution units and register files, as well as independent instruction buffers. In the blocks there are also SFU (Special Function Units) to handle transcendental and graphics interpolation instructions. [7]

The SMM is represented in Figure 3.

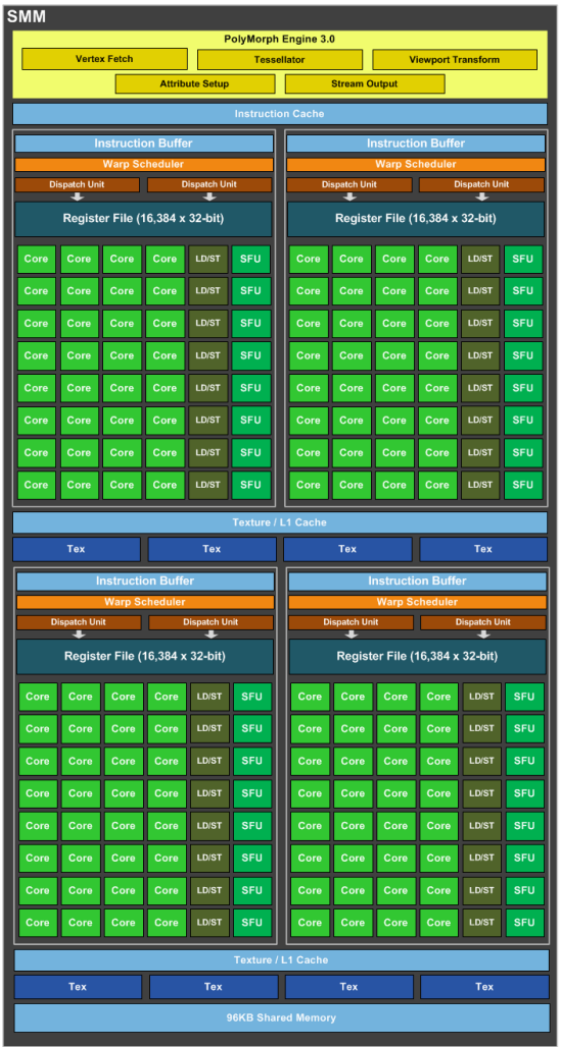

Figure 3: NVIDIA Maxwell SMM Architecture

Peripherals

[3] The Jetson Nano peripherals include:

- I2C - supports communication with multiple I²C Devices
- SPI - In master or slave mode, duplex synchronous and serial communication
- UART - controller provides serial data synchronization and data conversion
- PWM, USB and PCIe controllers
- I2S - transport streaming audio data between system memory and an audio codec
    - Also Supports PCM which is a standard method used to digitize audio (particularly voice) patterns for transmission over digital communication channels

Memory

[1] In terms of memory, Jetson Nano, contains two types: a LPDDR4 4GB, 64 bit RAM, which is Low-Power Double Data Rate, volatile and optimized for low power consumption and targeted for mobile computers and devices such as mobile phone. The other type of memory present in the Jetson Nano is persistent memory that is available only through the SD card. This memory type is very important since there is no internal storage in Jetson Nano, there is support for an SD card, for loading operating systems and persistent data storage.

Camera & Display

Has one MIPI CSI-2 DPHY lane. It implements a Camera Serial Interface receiver which receives data from an external camera module with CSI transmitter. Supports 1080p, 4K, 8K, low power, low electromagnetic interference. An example of CSI utilization with a camera can be found in Figure 4, where a camera was connected to the MIPI CSI-2 DPHY lane of the nano.

In terms of display the Jetson Nano also contains an HDMI output port as well as a DisplayPort Connector in order to connect the board to a display device that supports HDMI or DisplayPort interfaces (monitor, video, projector, etc). [1] [12]

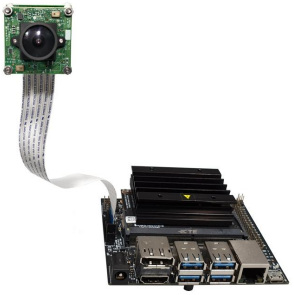

Figure 4: Example of a camera utilization with Jetson Nano

Connectivity

[1] In order to connect to outside devices or networks, the Jetson Nano contains one Gigabit Ethernet Port which is a transmission technology based on the Ethernet protocol capable of transmitting 1 billion bits per second (1 Gigabit) and 4 x USB3.0 ports.

In terms of wireless connectivity, the NVIDIA Jetson Nano does not contain a WiFi/Bluetooth module. However there is a connector M.2 Key E (specification for internal expansion cards on PC mainboards and notebook computers) located under the board that allows the integration with wireless network cards. Therefore in order to have wireless connectivity it is necessary to have a wireless card compatible with M.2 Key E and a set of radio antennas to support radio communications. [5] [4]

Figure 5: Wireless Card with M.2 Key E Connector and antennas

Power

Supplying power to the Jetson Nano can be done in several ways: through Micro USB connector or through a DC Barrel Jack. It is also possible to provide power via PoE using for example a splitter that “splits” a signal into power and data (Figure 6).

Figure 6: Splitter that splits ethernet signal into data (Ethernet) and power(DC Barrel Jack)

PoE can also be done by a PoE HAT, which is connected to the extra 4 pins that the Jetson contains directed specifically for PoE. Power can also be supplied via GPIO (in the 5V pins), connecting GPIO 5V pins to a 5V source and feeding energy directly to the board. However this can be dangerous because there is no regulation or fuse protection on the GPIO against over-voltage or current spikes. There are also HATs that can take care of this process in a safe manner. [6] [1]

Power Management

Jetson contains mechanisms and modules to implement a tiered structure of power and clock gating in a complex environment that optimizes power consumption:

- **Power Management Controller (PMC) -** Provides an interface to an external power managers and controls voltage transitions for the SoC
- **Real Time Clock (RTC) -** Maintains the ability to wake the system based on either a timer event or an external trigger
- **Clock and Power Gating -** The SoC aggressively employs clock and power gating to power-off modules which are idle
- **Dynamic Voltage and Frequency Scaling (DVFS) -** Raises voltages and clock frequencies when demand requires, lowers them when less is sufficient, and removes them when none is needed

The Jetson module operates in three main power modes: **OFF** - System is not powered, therefore the Jetson is not in operation, **ON** - Jetson module is fully functional and operates normally and **SLEEP** - allows the Jetson module to quickly resume to an operational state

without performing a full boot sequence. Operates in low power with enough circuitry powered to allow the device to resume and re-enter the ON state. [3]

Video

Supports two codecs for video codification: H.265 (HEVC - High Efficiency Video Coding), H.264 (AVC - Advanced Video Coding).

Video decoding formats include: 4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 18x 720p @ 30. Video encoding formats:4K @ 30 | 4x 1080p @ 30 | 9x 720p @ 30. [1]

GPIO

[13] GPIO contains a 40 Pin Header whose functions and names are represented in Figure7.

Sysfs	Name	Pin	Pin	Name	Sysfs
	3.3V DC	1	2	5V DC	
	I2C_2_SDA	3	4	5V DC	
	I2C_2_SCL	5	6	GND	
gpio216	AUDIO_MCLK	7	8	UART_2_TX	
	GND	9	10	UART_2_RX	
gpio50	UART_2_RTS	11	12	I2S_4_CLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3V DC	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_2_CS0	gpio19
	GND	25	26	SPI_2_CS1	gpio20
	IC2_1_SDA	27	28	I2C_1_SCL	
gpio149	CAM_AF_EN	29	30	GND	
gpio200	GPIO_PZO	31	32	LCD_BL_PWM	gpio168
gpio38	GPIO_PE6	33	34	GND	
gpio76	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
gpio12	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio77
	GND	39	40	I2S_4_SDOUT	gpio78

Figure 7: GPIO of the Jetson Nano

NVIDIA JetPack SDK

JetPack SDK provides a full development environment for hardware-accelerated AI-at-the-edge development. It includes Jetson Linux Driver Package with bootloader, Linux kernel, **Ubuntu desktop environment**, and a complete set of **libraries** for acceleration of

GPU computing, multimedia, graphics, and computer vision. It also includes **samples**, **documentation**, and **developer tools** for both host computer and developer kit. [8]

OS image

The official operating system for the Jetson Nano is the **Linux4Tegra**, based on **Ubuntu 18.04**. This is available via the included SD card image, which is **designed to run NVIDIA hardware**. But Jetson Linux 34.1 is **a developer preview release** which is useful for deploying computer vision and deep learning. It includes:

- Linux Kernel 5.10
- UEFI based bootloader
- Ubuntu 20.04 based root file system
- NVIDIA drivers
- necessary firmwares
- toolchain

Libraries

- TensorRT and cuDNN for high-performance deep learning applications.
    - TensorRT is a high-performance deep learning inference runtime for image classification, segmentation, and object detection neural networks.
    - CUDA Deep Neural Network library provides high-performance primitives for deep learning frameworks.
- CUDA for GPU accelerated applications across multiple domains.
- NVIDIA Container Runtime for containerized GPU accelerated applications.
    - The container runtime enables creation, distribution, and use of containerized GPU accelerated applications.
    - Jetson Multimedia API package provides low level APIs for flexible application development.
- Camera application API: libargus offers a low-level frame-synchronous API for camera applications, with per frame camera parameter control, multiple (including synchronized) camera support, and EGL stream outputs. RAW output CSI cameras needing ISP can be used with either libargus or GStreamer plugin. In either case, the V4L2 media-controller sensor driver API is used.
- Sensor driver API: V4L2 framework enables video decode, encode, format conversion and scaling functionality.

- VPI and OpenCV, and for visual computing applications.
- OpenCV is the leading open source library for computer vision, image processing and machine learning, and now features GPU acceleration for real-time operation.
- VPI (Vision Programming Interface), a software library that provides Computer Vision / Image Processing algorithms implemented on PVA1 (Programmable Vision Accelerator), GPU and CPU.

Samples

There are a number of samples demonstrating use of JetPack components. These are included in the reference filesystem, and can be compiled on the developer kit.

JetPack component	Sample locations on reference filesystem
TensorRT	/usr/src/tensorrt/samples/
cuDNN	/usr/src/cudnn_samples_<version>/
CUDA	/usr/local/cuda-<version>/samples/
MM API	/usr/src/jetson_multimedia_api
OpenCV	/usr/share/opencv4/samples/
VPI	/opt/nvidia/vpi/samples/

Developer Tools

Tools for application development and debugging include:

- Nsight Eclipse Edition extension in Eclipse for development of GPU accelerated applications
- CUDA-GDB for application debugging
- CUDA-MEMCHECK for debugging application memory errors

Tools for application profiling and optimization:

- Nsight Systems and nvprof for application profiling across GPU and CPU.
- Nsight Graphics for graphics application debugging and profiling
- NVIDIA Nsight Compute for interactive CUDA kernel profiling
- NVIDIA Nsight Compute (bundled with CUDA Toolkit) is an interactive kernel profiler for CUDA applications.

Install OS

To install jetson nano oficial OS we should burn JetPack SDK image to usb pen using etcher and on the first boot we must finish installation. [11]

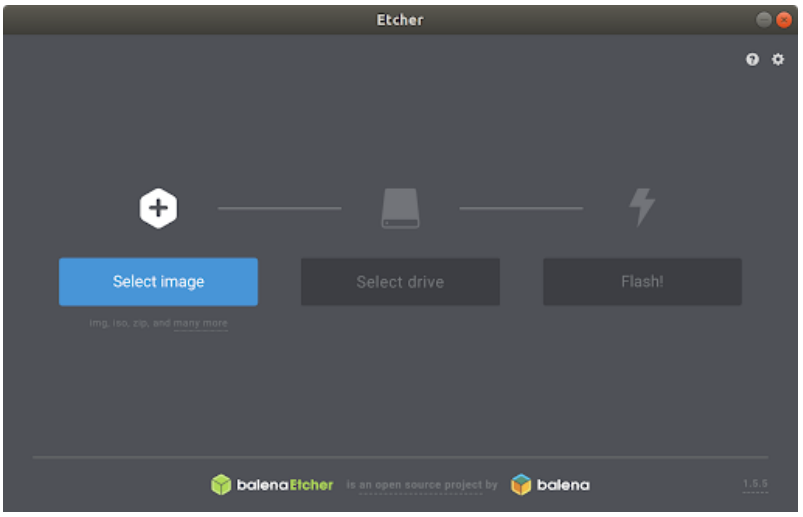

Figure 8: Etecher is a software to burn OS images to general peripherals like usbs and SD cards

Ways to boot/connect to jetson nano

Boot/connecting to the Jetson Nano can be done by using a display monitor with usb keyboard and mouse or through a headless mode with a bash command screen which emulates jetson terminal through a physical serial port. It is also possible to use VNC (virtual network computing) to have remote access to the graphical environment

Support for Upgrades

As it is a linux distribution system and has a Ubuntu based file root system we can upgrade and install new software through the most common package control system APT(Advanced Packaging Tool). It's possible to do:

sudo apt update - updates that list of software packages

sudo apt upgrade - upgrade those packages

APT also allows to install and uninstall specific packages

Support for Remote Access

As jatson nano is a Ubuntu based system and it's close to being a general purpose computer we can remote access a computer with ssh, Graphical desktop environment with VNC and transfer files with secure copy and FTP.

Jetson nano 2G Developer kit

The official NVIDIA Jetson nano 2G Developer kit costs about 99$ and it includes:

- NVIDIA Jetson module and reference carrier board
- Quick Start / Support Guide
- 802.11ac wireless adapter and extension cable

it doesn't include:

- microSD Card
- keyboard and mouse
- HDMI display cale
- USB-C power supply

Software Architecture

The following diagram (Figure 9) shows the NVIDIA Jetson Linux architecture. [10]

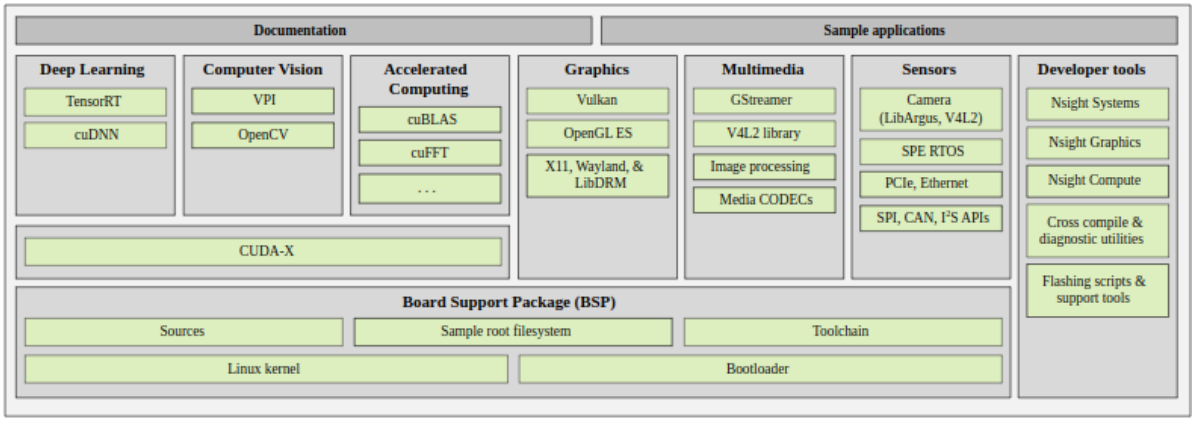

Figure 9: Software Architecture

Community Projects

There are a ton of community projects in which people can submit and publish their projects in the NVIDIA website, and all have their source code public. [9]

Example of some projects are:

- AI whiteboard
- real-time human pose estimation
- handwriting ml classifier
- dart score detector

Bibliography

- [1] - NVIDIA Jetson Nano Developer kit
  - https://developer.nvidia.com/embedded/jetson-nano-developer-kit
- [2] - ARM® Cortex® -A57 MPCore™ Processor, Technical Reference Manual
  - https://documentation-service.arm.com/static/5e906864c8052b16087605a4
- [3] - Jetson Nano DataSheet
  - https://developer.nvidia.com/embedded/dlc/jetson-nano-system-module-datasheet
- [4] - The M.2 Interface
  - https://www.delock.de/infothek/M.2/M.2_e.html
- [5] - Jetson Nano + Intel Wifi
  - https://jetsonhacks.com/2019/04/08/jetson-nano-intel-wifi-and-bluetooth/
- [6] - Use The Power - Jetson Nano
  - https://www.youtube.com/watch?v=jq1OqBe267A
- [7] - Maxwell: The Most Advanced CUDA GPU Ever Made
  - https://developer.nvidia.com/blog/maxwell-most-advanced-cuda-gpu-ever-made/
- [8] - NVIDIA JETPACK
  - https://docs.nvidia.com/jetson/jetpack/introduction/index.html
- [9] - jetson Community Projects
  - https://developer.nvidia.com/embedded/community/jetson-projects
- [10] - software architecture
  - https://docs.nvidia.com/jetson/archives/r34.1/DeveloperGuide/text/AR/JetsonSoftwareArchitecture.html
- [11] - install OS
  - https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write
- [12] - MIPI Camera Serial Interface 2
  - https://www.mipi.org/specifications/csi-2
- [13] - How to use GPIO Pins in NVIDIA Jetson Nano Kit
  - https://maker.pro/nvidia-jetson/tutorial/how-to-use-gpio-pins-on-jetson-nano-developer-kit