Embedded Basic Knowledge

A small introduction to the embedded computing world

Who am I? Alexjan Carraturo

Embedded System

"An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today."

https://en.wikipedia.org/wiki/Embedded_system

Embedded Systems

- **PLC (Programmable Logic Controller)**
  - Industrial, easy programmable, block design, hard real-time constrains
- **MCU (MicroController Unit)**
  - Small system on chip, with CPU, programmable I/0, DSP, ADC, DAC, commonly low memory, low storage, low speed and low power. OS-less or RealTime-OS
- **SoC (System on Chip)**
  - Single or multicore CPU, often GPU, WiFi, networking, audio/video decoders. Higher speed and performance, used in Mobile, HMI, Automotive (info), vision.

ARCH

ARM®

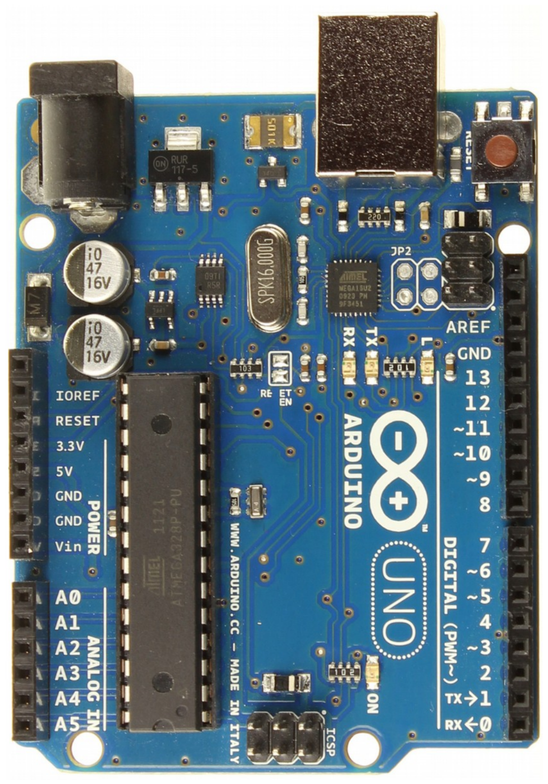
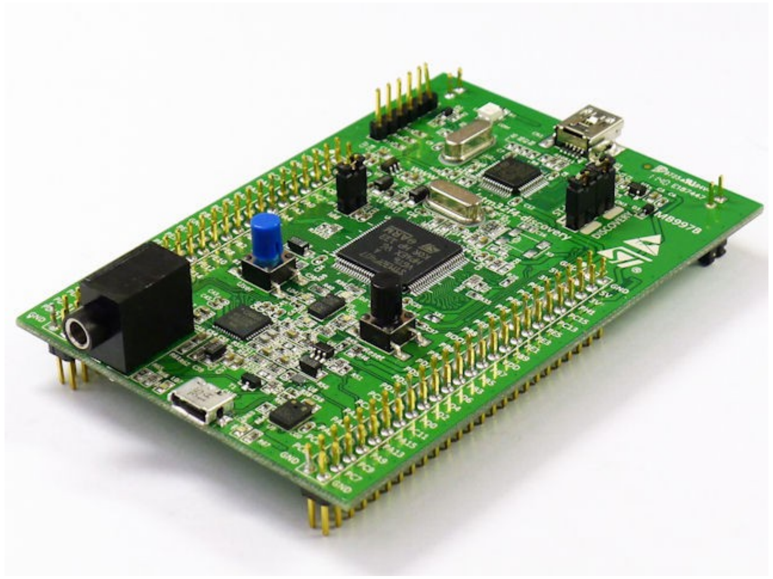
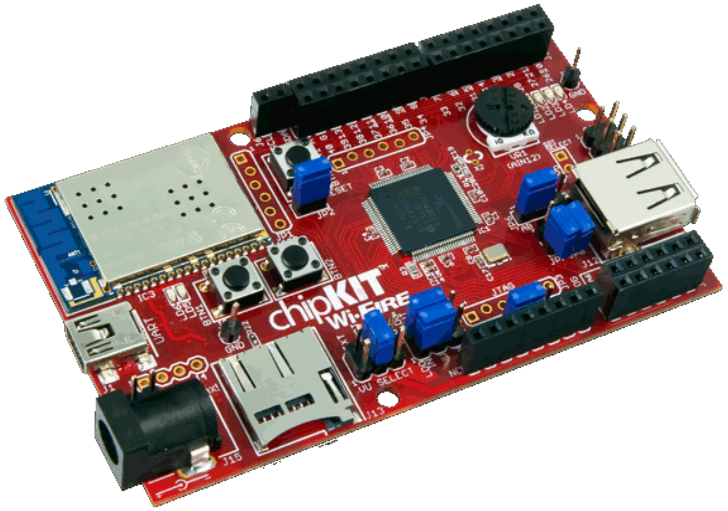
ATMEL

intel®

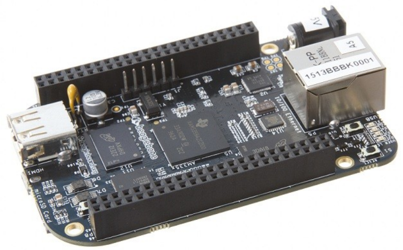
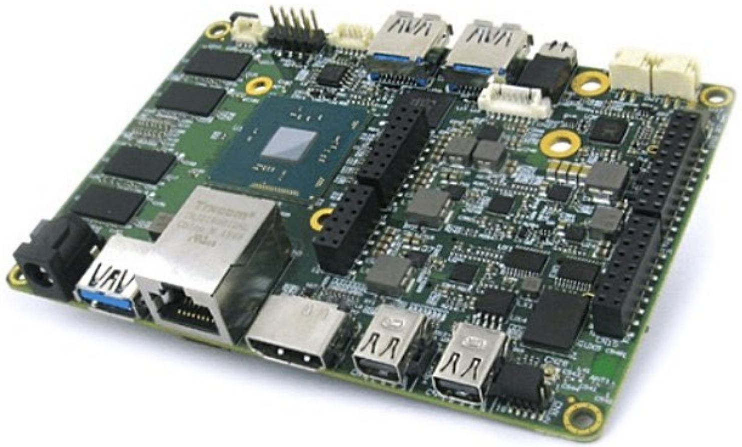
MIPS

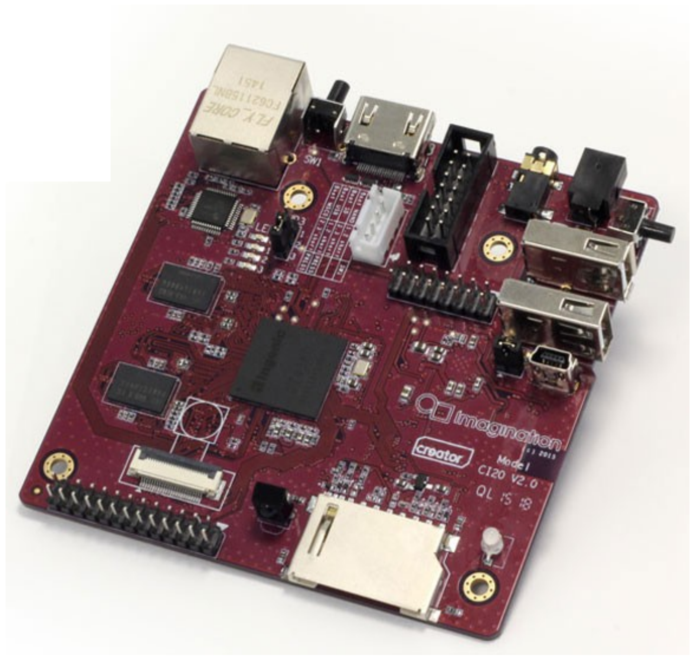
by Imagination

MCU

SoC – Evaluation Board



Architecture, Core, SoC

- SoC != CPU
- SoC != Core
- Architecture and ISA

- Es: Cortex
  - ✗ Cortex-A (Application System on Chip)
  - ✗ Cortex-M (Micro controller Unit)
  - ✗ Cortex-R (Real time application)
- Es. MIPS
  - ✗ Profile P (Performance)
  - ✗ Profile I (Intermediate)
  - ✗ Profile M (Micro)

IP

- GPU
- DSP/ISP
- Encorder/Decoder
- Connectivity (Wifi, Ethernet, Bluetooth, 6LowPan GPS…)
- I/O Controller (Serial, SPI, I2S, I2C, GPIO…)
- Special purpose device

Buzzwords and Underrated

- SoC Brand
- GHz
- Nr. Core
- RAM (???)
- Storage
- Nr. Bit

- I/O
- How many?
- Quality
- RAM & Bus Speed
- IP's on SoC

SoC - Eval Comparative

	Raspberry PI3-B	ODROID C2	Imgtec Ci20
Arch	Quad ARM Cortex A53	Quad ARM Cortex A53	Dual Xburst (MIPS32r2)
Freq	1.2 GHz	1.5 GHz	1.2 GHz
Bit	64	64	32
Ram	1Gb LPDDR2	2 Gb DDR3	1 Gb DDR3
Storage	MicroSD	EMMc + MicroSD	EMMc + MicroSD
GPU	Broadcom VideoCore 4	ARM Mali 450	IMG SGX 540
Cost	35$	40$	50£

Eval boards : use cases

- Low Power Desktop
- Mediacenter
- Smarthome (Home Automation)
- Surveillance
- Small server
- Network Gateway
- Simply I/O control (Led, thermo, power…)
- Experiments (no limit)

Run a full Embedded Linux System

- **Bootloader**
  - ➔ Pre-bootloader and environment (likely pre-built)
- **Kernel**
  - ➔ Pre-build, Vanilla or platform specific
- **Rootfs**
  - ➔ Pre-build Linux Distribuion or custom build

Development Environment Basic

- **Custom Cross-toolchain**
  - Gcc, glibc, binutils
  - Gdb (optional)

Pro: Maximum level of optimization

Con: long build time, difficult procedure, unique environment

- **Pre-built toolchain**
  - Linaro (ARM)
  - Codescape (MIPS)

Pro: Shared environment (tested), easy to get

Con: Generic per architecture, not for all ARCH.

Bootloader

Bootloaders

- Das U-Boot (most likely)
- Redboot
- Barebox
- Little-kernel (mostly android)

Few and uncommon situations ask for a rebuild of the bootloader.

More likely would be required to adapt environment variable (change boot device and boot parameter). Bootloader images and source are usually available by the board producer.

Linux Kernel

Using a pre-compiled one

- Pre-built images of Distro
- Pre-installed images

Build an image

- kernel source (vanilla or custom)
- defconfig
- device tree
- external driver

ROOTFS (Pre-built)

- **Pre-installed images**
  - Usually full working but limited
- **Common GNU/Linux Distribution**
  - Debian (all)
  - Gentoo (all*)
  - OpenSUSE (ARM)
  - Fedora (ARM, MIPS)
  - Ubuntu (ARM)

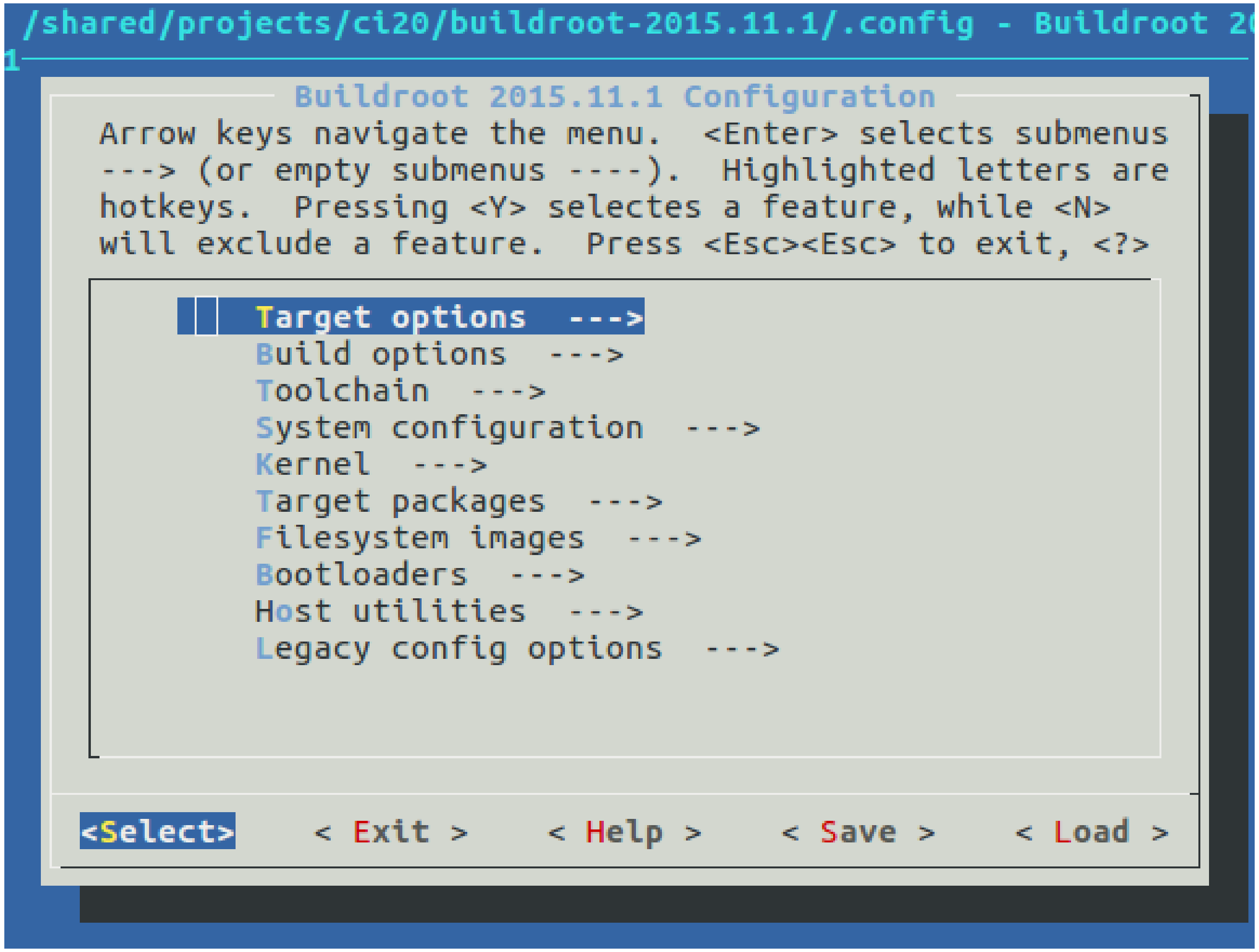
* Could require human sacrifices

ROOTFS (Build from source)

- Buildroot
- OpenWRT (networking)
- openEmbedded
- Yocto
- Android (AOSP)
- Tizen
- LFS (Linux From Scratch)

In increasing order of requested desperation

Buildroot

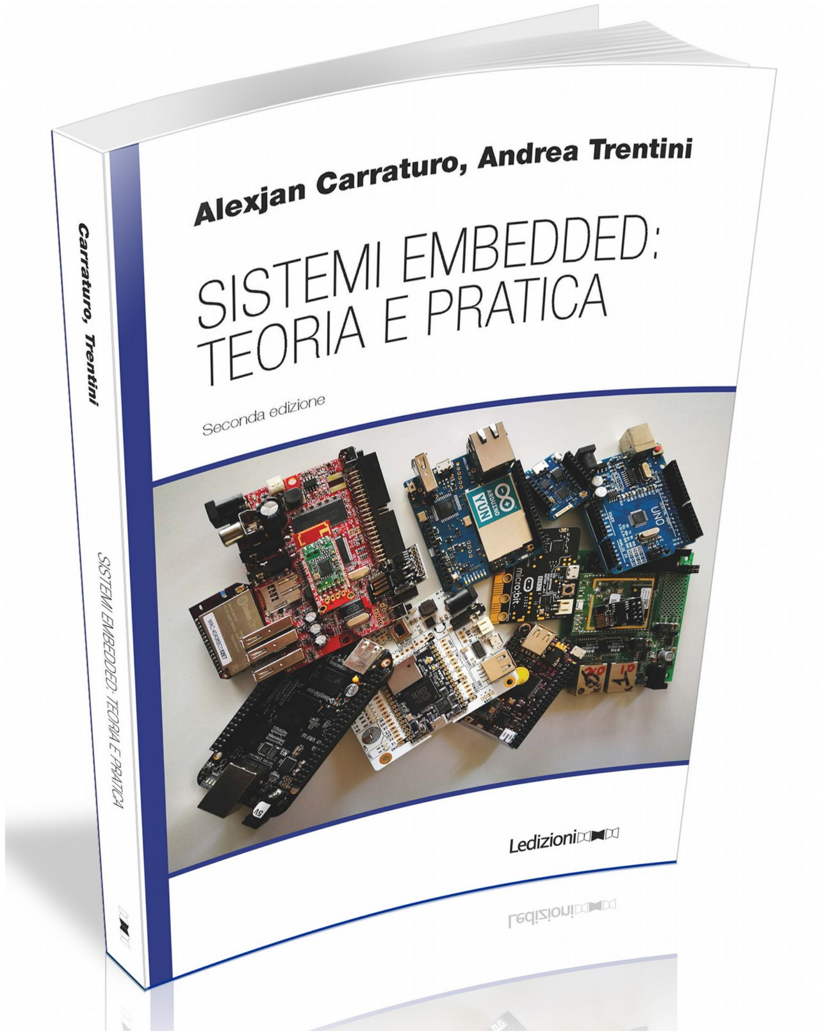


```
/shared/projects/ci20/buildroot-2015.11.1/.config - Buildroot 20
1
                  Buildroot 2015.11.1 Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus
  ---> (or empty submenus ----).  Highlighted letters are
  hotkeys.  Pressing <Y> selectes a feature, while <N>
  will exclude a feature.  Press <Esc><Esc> to exit, <?>

            Target options  --->
            Build options  --->
            Toolchain  --->
            System configuration  --->
            Kernel  --->
            Target packages  --->
            Filesystem images  --->
            Bootloaders  --->
            Host utilities  --->
            Legacy config options  --->

 <Select>    < Exit >    < Help >    < Save >    < Load >
```

Buildroot

- **Buildroot**
  - Cross toolchain (from scratch or prebuilt)
  - Kernel (with defconfig)
  - Bootloader (few platform)
  - Target Packages
    - Multimedia
    - Networking
    - Basic Desktop
    - Tool
    - Debugging, tuning and benchmark
    - ...

Book


Alexjan Carraturo, Andrea Trentini
SISTEMI EMBEDDED: TEORIA E PRATICA
Seconda edizione
Ledizioni
Carraturo, Trentini
SISTEMI EMBEDDED: TEORIA E PRATICA

Book

- Cap 1, “Introduzione”
- Cap 2, “Concetti generali”: i mattoni di base legati al mondo dell’informatica, della programmazione e de sistemi operativi;
- Cap 3, “Richiami di elettronica”: fornisce le conoscenze minime per capire l’interfacciamento elettrico tra un sistema embedded e il mondo fisico;
- Cap 4, “Architetture Embedded”: panoramica delle piattaforme embedded più diffuse;
- Cap 5, “Memorie, I/O e comunicazione”: panoramica sulle tecnologie di memorie e comunicazione dei sistemi embedded.
- Cap 6, “Il sistema operativo”: approfondimento sui sistemi embedded di fascia “alta”, con particolare riferimento a GNU/Linux;
- Cap 7, “Configurazione GNU/Linux”: preparazione di un sistema operativo (basato su GNU/Linux) da installare su una piattaforma embedded;
- Cap 8, “FreeRTOS”: approfondimento sul diffusissimo “sistema operativo” per piattaforme embedded real-time;
- Cap 9, “Arduino e Wiring”: approfondimento su una piattaforma embedded di fascia “bassa”, senza sistema operativo, è stata scelta la più diffusa attualmente: Arduino;
- Cap 10, “Rete e protocolli”: panoramica sui protocolli di rete e introduzione ad alcuni protocolli di comunicazionedi uso diffuso nei sistemi embedded;
- App A, “Ambiente di Test”: preparazione di un ambiente di sviluppo/testing basato sull’interazione fra board e workstation;
- App B, “Esempi pratici”: alcuni esempi pratici di programmazione ed uso di comuni sistemi embedded.

Book

http://sistemiembedded.cc/

https://www.amazon.it/dp/8867059432/ref=cm_sw_em_r_mt_dp_U_J0KMCbEGKHG4X

https://www.ledizioni.it/prodotto/a-carraturo-a-trentini-sistemi-embedded-teoria-pratica/

Thank you

alexjan.carraturo@gmail.com