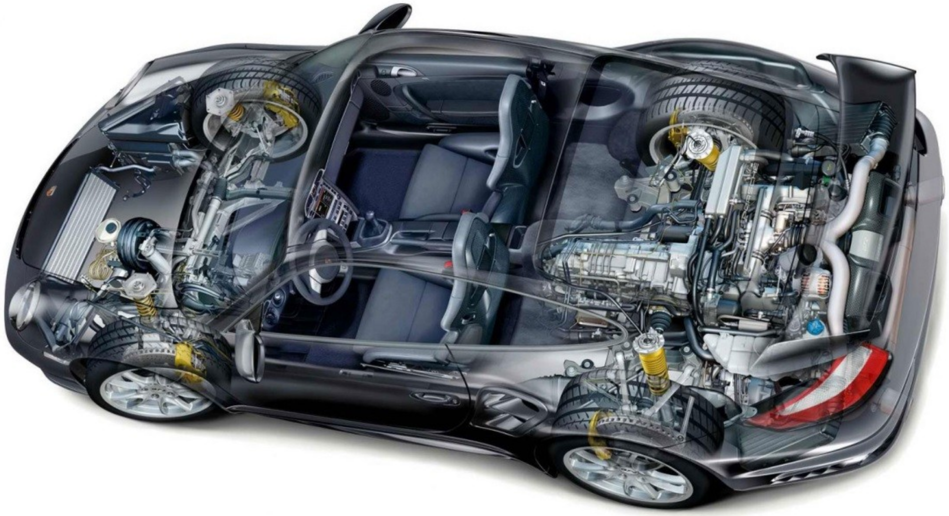
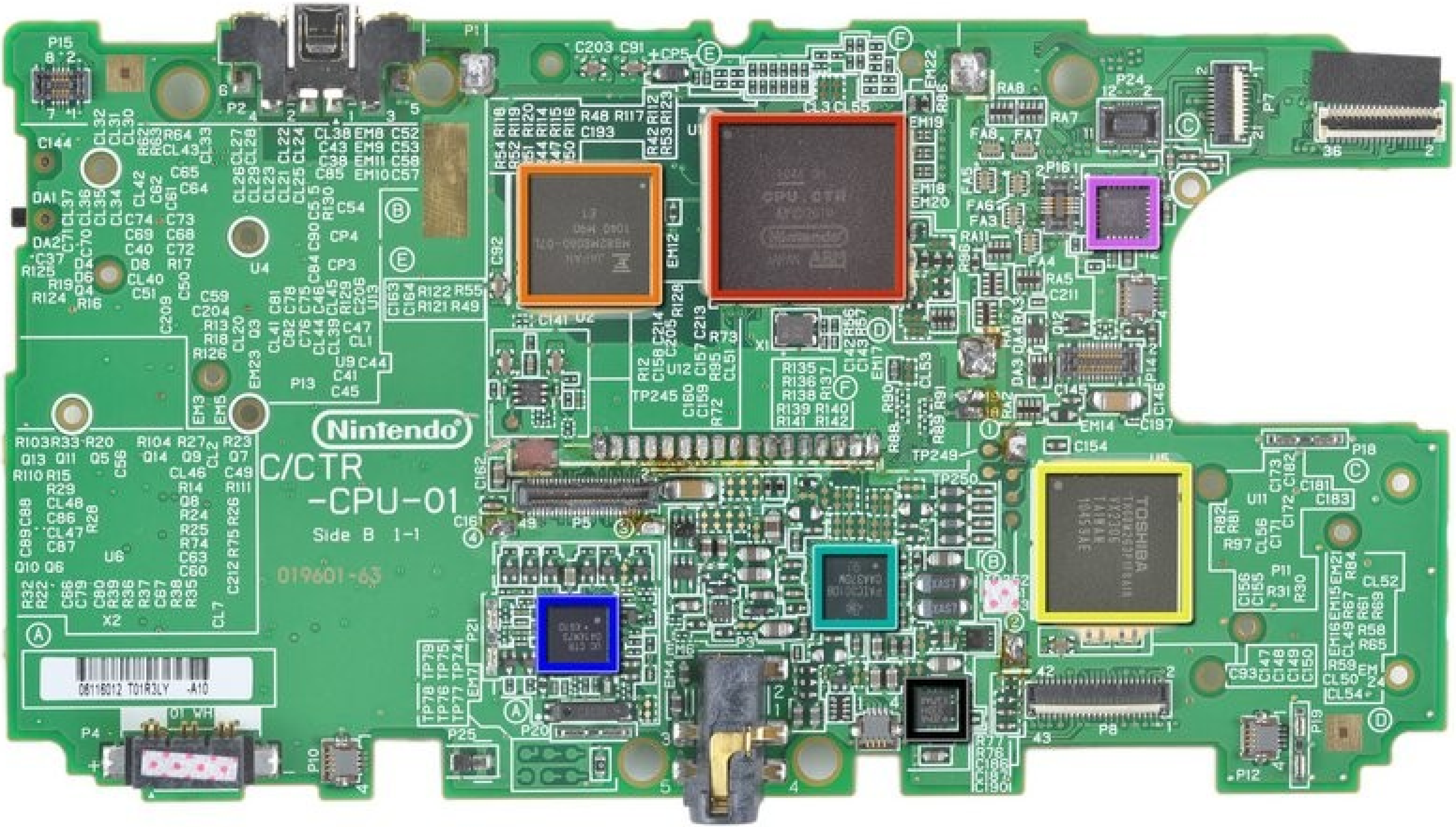
Embedded Systems Engineering



Focus on Cyber-Physical Systems

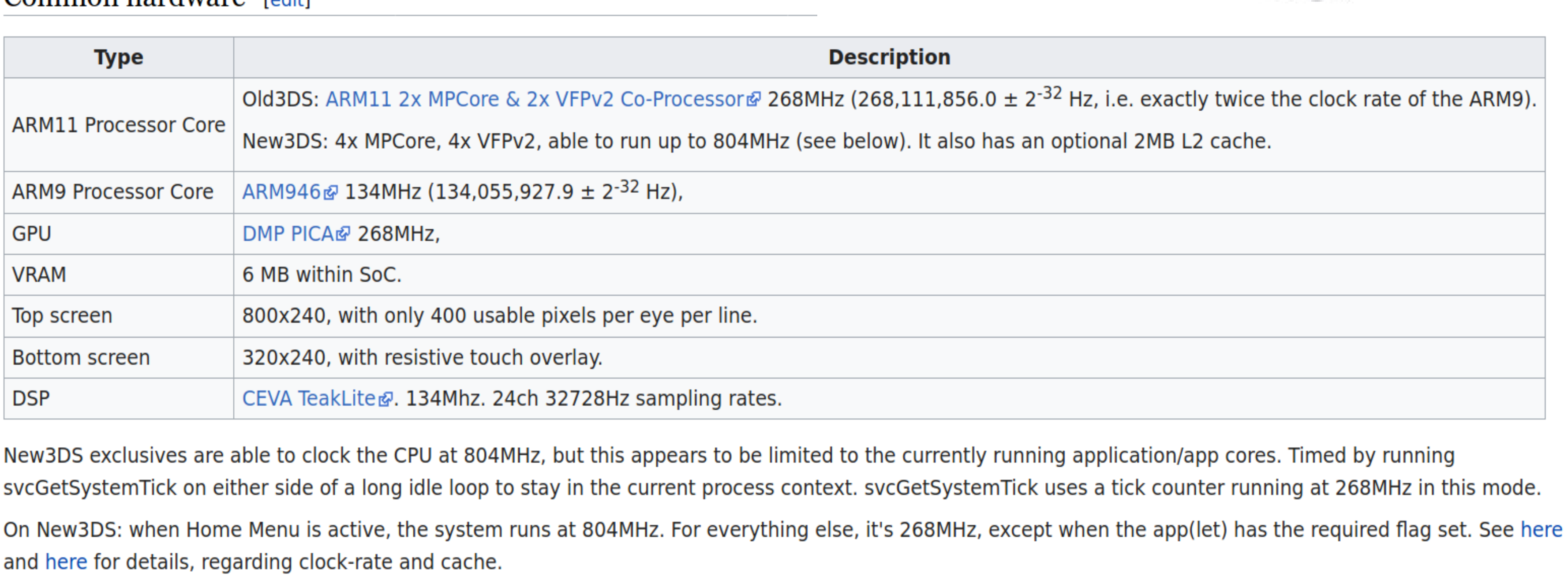
- Eachine H8 mini
- ARM assembler + C
- QEMU-System-arm
- C optimisation
- Compiler and linker
- STM32F4 Board
- Firmware
- Technical Reference Manual
- STM32Cube
- State machines
- verilog
- Lexical Analysis
- Parsing
- Abstract Syntax Tree
- Clang/LLVM
- Clang tools
- FPGAs – software synthesis
- Vivado Design suite

Nintendo 3DS board



Nintendo 3ds Hardware

- https://3dbrew.org/wiki/Hardware
-



Common hardware [edit]

Type	Description
ARM11 Processor Core	Old3DS: ARM11 2x MPCore & 2x VFPv2 Co-Processor 268MHz (268,111,856.0 ± 2^{-32} Hz, i.e. exactly twice the clock rate of the ARM9). New3DS: 4x MPCore, 4x VFPv2, able to run up to 804MHz (see below). It also has an optional 2MB L2 cache.
ARM9 Processor Core	ARM946 134MHz (134,055,927.9 ± 2^{-32} Hz),
GPU	DMP PICA 268MHz,
VRAM	6 MB within SoC.
Top screen	800x240, with only 400 usable pixels per eye per line.
Bottom screen	320x240, with resistive touch overlay.
DSP	CEVA TeakLite. 134Mhz. 24ch 32728Hz sampling rates.

New3DS exclusives are able to clock the CPU at 804MHz, but this appears to be limited to the currently running application/app cores. Timed by running svcGetSystemTick on either side of a long idle loop to stay in the current process context. svcGetSystemTick uses a tick counter running at 268MHz in this mode.

On New3DS: when Home Menu is active, the system runs at 804MHz. For everything else, it's 268MHz, except when the app(let) has the required flag set. See here and here for details, regarding clock-rate and cache.

Game Boy Advance GBA

The memory layout of the GBA

https://kuleuven-diepenbeek.github.io/cpp-course/gba-in-c/labo-3/

A clear view of I / O addresses and their function is important. Addresses fall in a range depending on the size of each system's memory. Below is a brief overview (source):

0x00000000 - 0x00003FFF - 16 KB System ROM (executable, but not readable)
0x02000000 - 0x02030000 - 256 KB EWRAM (general purpose RAM external to the CPU)
0x03000000 - 0x03007FFF - 32 KB IWRAM (general purpose RAM internal to the CPU)
0x04000000 - 0x040003FF - I/O Registers
0x05000000 - 0x050003FF - 1 KB Colour Palette RAM
0x06000000 - 0x06017FFF - 96 KB VRAM (Video RAM)
0x07000000 - 0x070003FF - 1 KB OAM RAM (Object Attribute Memory — discussed later)
0x08000000 - 0x???????? - Game Pak ROM (0 to 32 MB)
0x0E000000 - 0x???????? - Game Pak RAM

Display setup GBA

There are 6 different “Video Modes” available that you have to turn on or off before you can draw anything on the screen. The GBA supports tilesets to draw sprites more efficiently (the 3 last modes), but for the time being we have enough to set the color pixel by pixel (the 3 first modes). The simplest mode without buffering is video mode 3 . This has a resolution of 240x160. Each pixel RGB values to address.

Besides mode 3 we also have to choose a “Background mode”. There are 4 background layers available that make it possible to create a 3D illusion by drawing layer by layer. BG mode 2 is sufficient for now.

Display setup GBA

```
#define MODE3 0x0003
#define BG2 0x0400

volatile unsigned int *display_control = (volatile unsigned int*) 0x4000000;

int main() {
    *display_control = MODE3 | BG2;
}
```

The video parameters are written to control register 0x4000000. We can form a combination of BG2 and Mode3 together with a bitwise operator , but you can also manipulate the bits separately. To clarify the cryptic registers we use preprocessor defines.

Nintendo GameBoy Advance (GBA)

The CPU is a 32-bit ARM7tdmi chip running at 16.78 MHz.

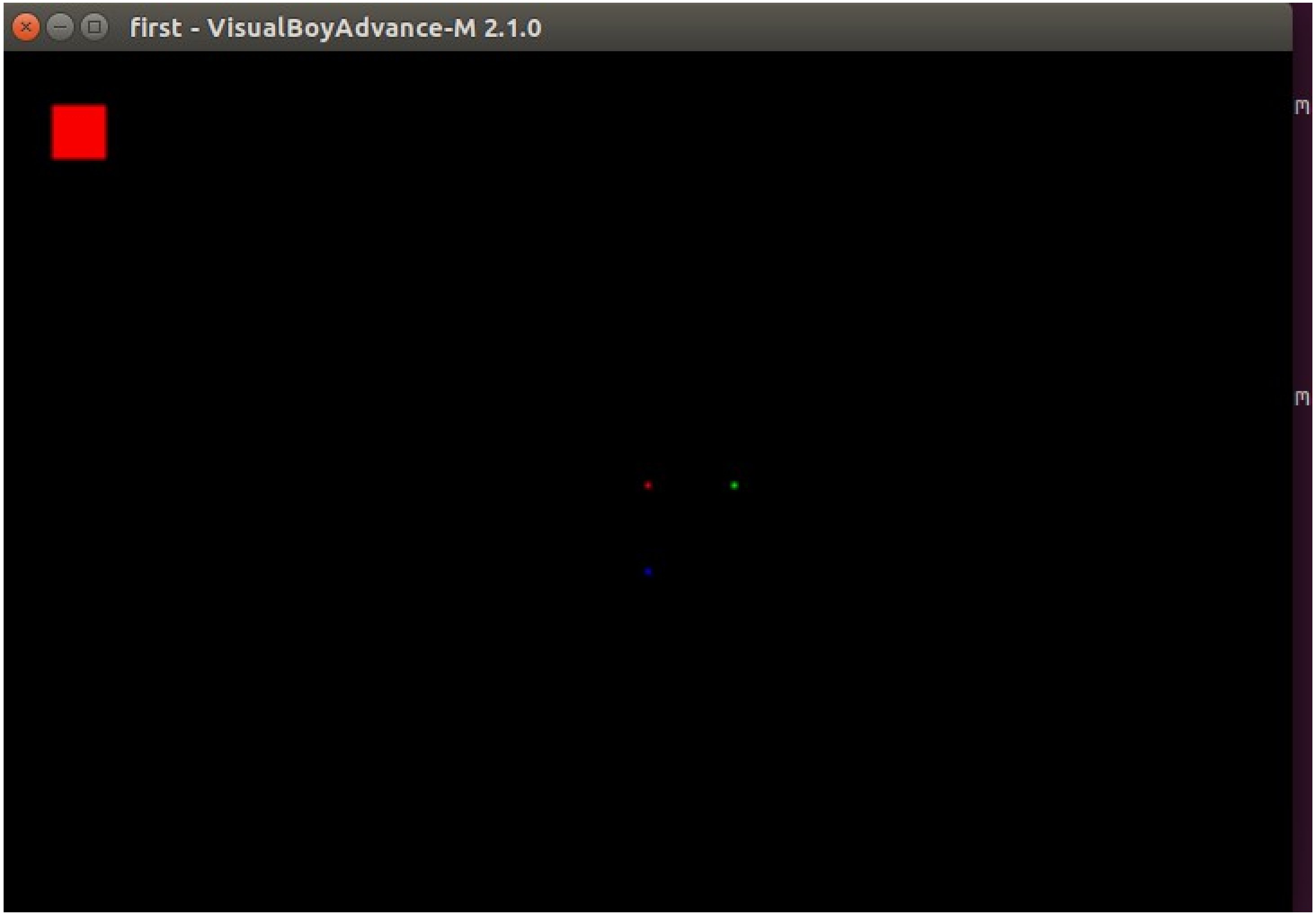
```
// 16 bit colour
// 5 bits red | 5 bits green| 6 bits blue
#define RGB(r,g,b) (unsigned short)(r + (g << 5) + (b << 10))
// First demo.

int main()
{
   *(unsigned int*)0x04000000 = 0x0403;

   ((unsigned short*)0x06000000)[120+80*240] = 0x001F;
   ((unsigned short*)0x06000000)[136+80*240] = 0x03E0;
   ((unsigned short*)0x06000000)[120+96*240] = 0x7C00;

   while(1);

   return 0;
}
```

Nintendo GameBoy Advance (GBA)

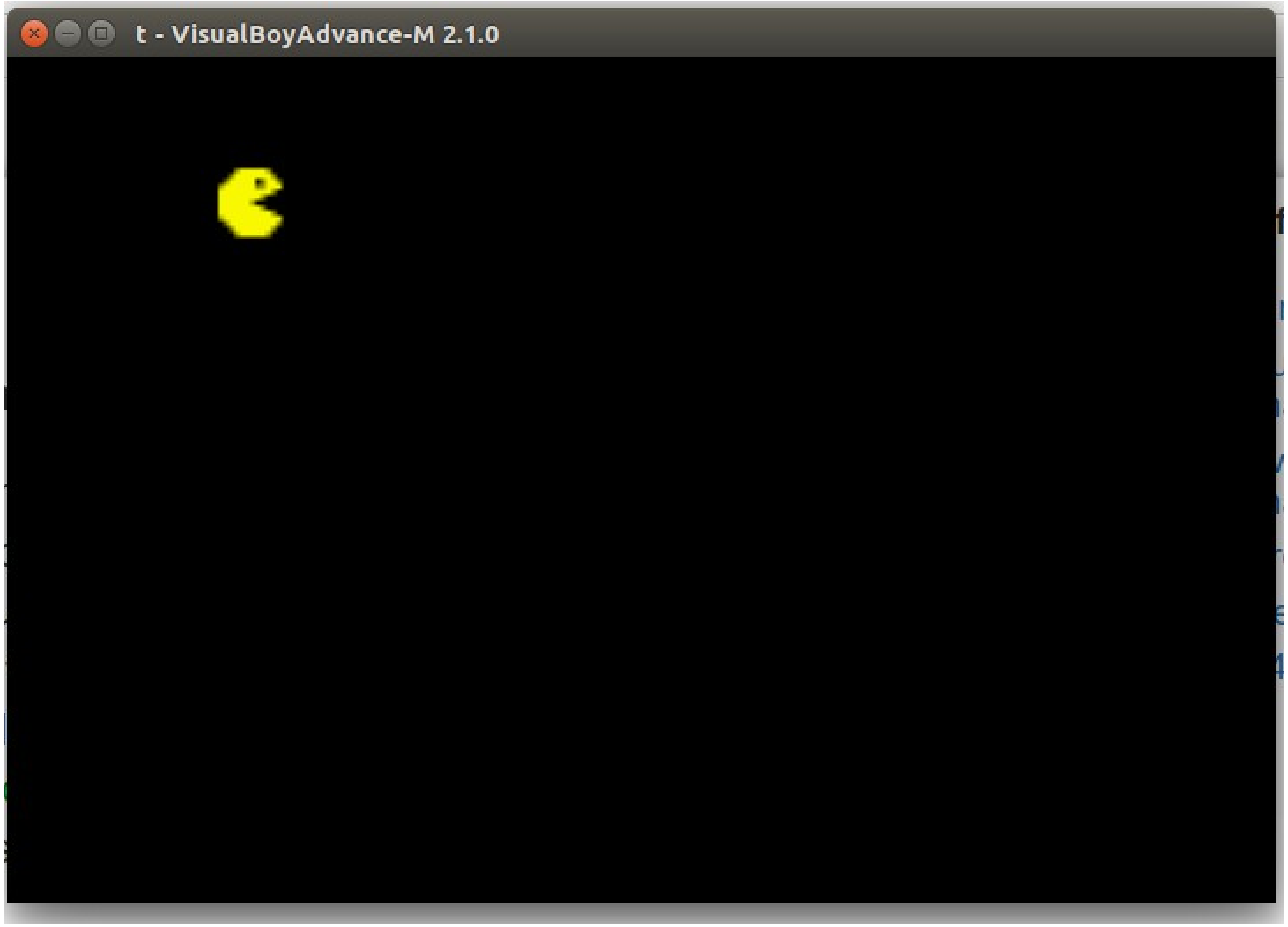
Draw a red square

```
// 16 bit colour
// 5 bits red | 5 bits green| 6 bits blue
 #define RGB(r,g,b) (unsigned short)(r + (g << 5) + (b << 10))

int main()
{
   *(unsigned int*)0x04000000 = 0x0403;
   ((unsigned short*)0x06000000)[120+80*240] = 0x001F;
   ((unsigned short*)0x06000000)[136+80*240] = 0x03E0;
   ((unsigned short*)0x06000000)[120+96*240] = 0x7C00;
   for (int j=10;j<20;j++)
      for (int i=10;i<20;i++)
          ((unsigned short*)0x06000000)[i+j*240] = 0x001F;

   while(1);

   return 0;
}
```

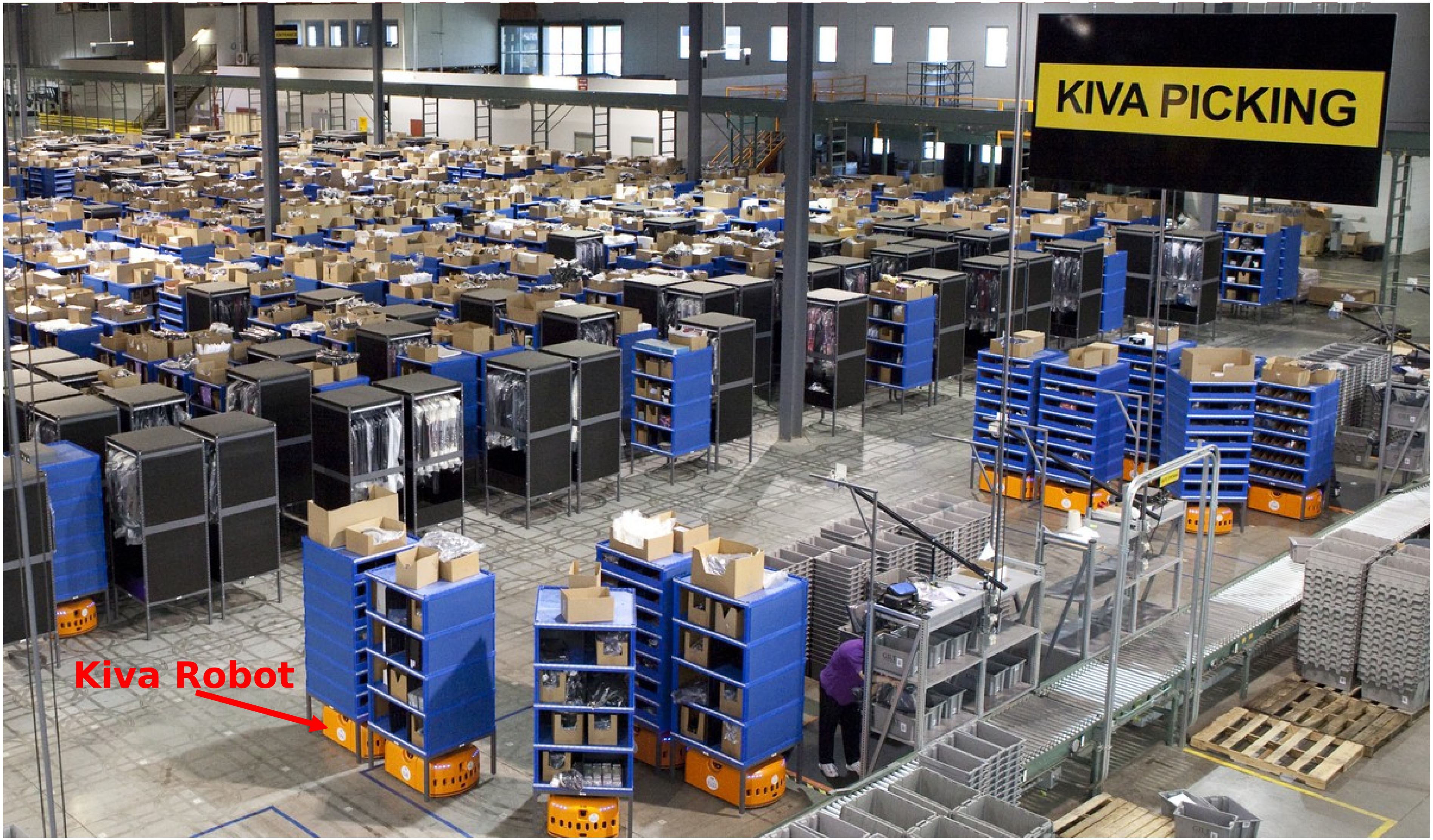
Nintendo GameBoy Advance (GBA)

- 

Packman on a GBA emulator

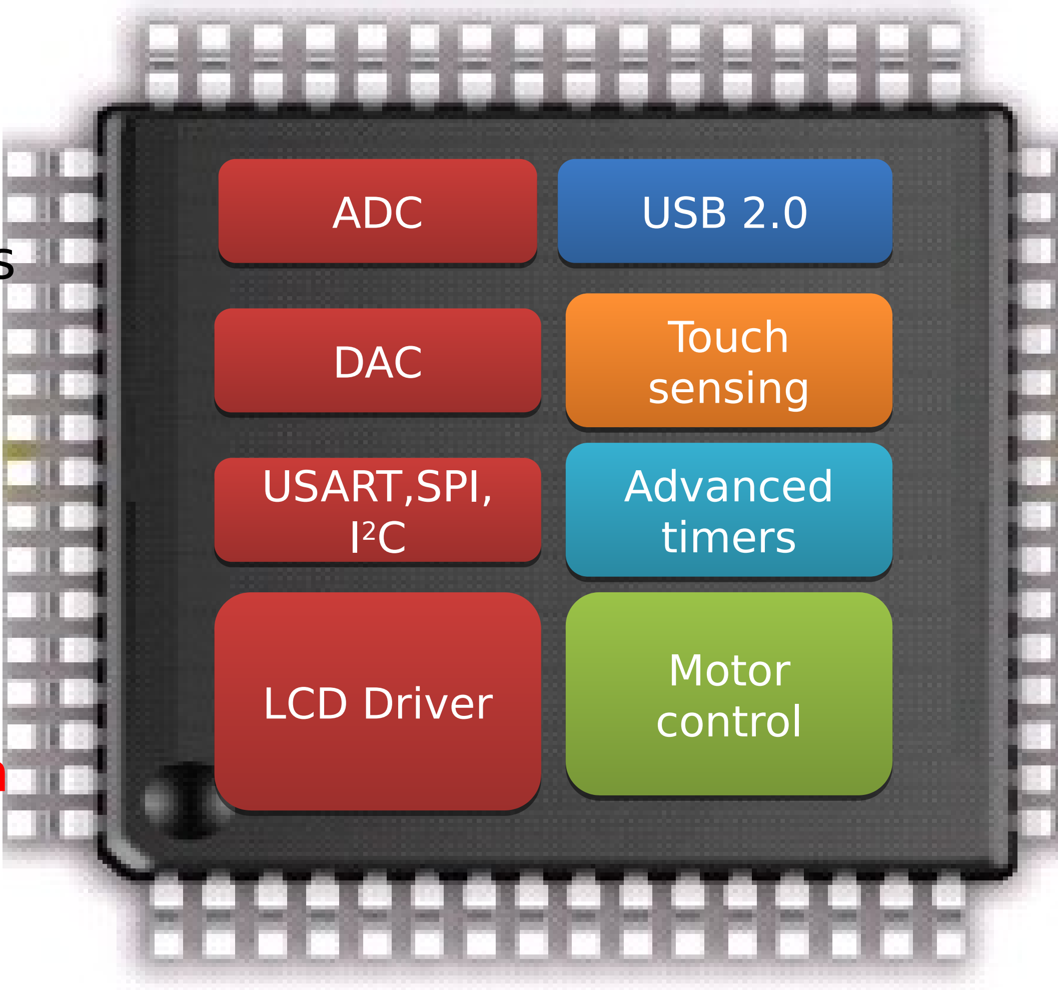


Amazon Warehouse

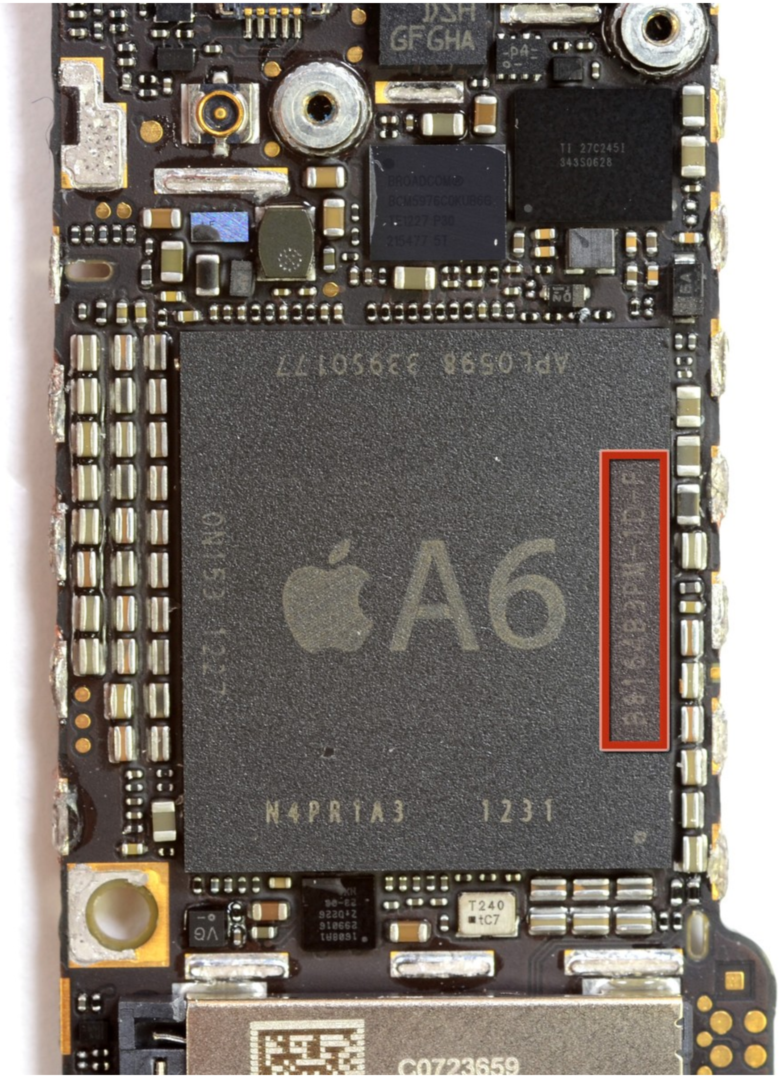


Why ARM processor

- As of 2005, **98%** of the more than one billion mobile phones sold each year used ARM processors
- As of 2009, ARM processors accounted for approximately **90%** of all embedded 32-bit RISC processors
- In 2010 alone, **6.1 billion** ARM-based processor, representing **95%** of smartphones, **35%** of digital televisions and set-top boxes and **10%** of mobile computers



iPhone 5 Teardown



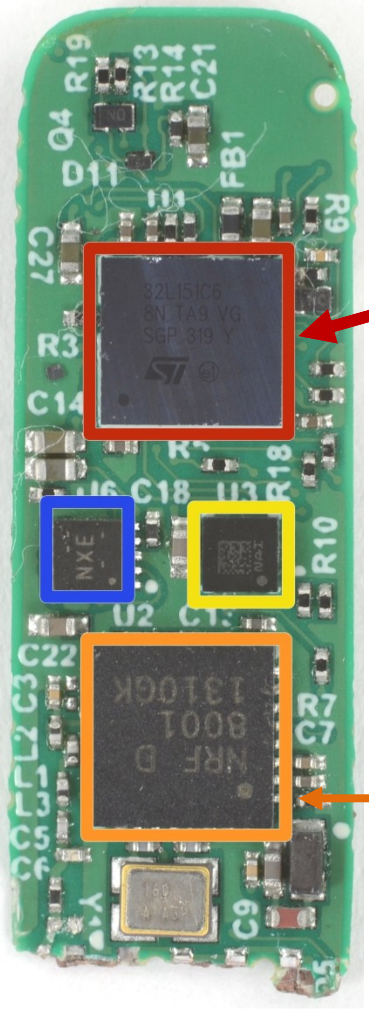
The A6 processor is the first Apple System-on-Chip (SoC) to use a custom design, based off the **ARMv7** instruction set.

http://www.ifixit.com

Apple Watch

- Apple S1 Processor
  - **32-bit ARMv7-A** compatible
  - # of Cores: **1**
  - CMOS Technology: 28 nm
  - L1 cache 32 KB data
  - L2 cache 256 KB
  - GPU PowerVR SGX543

Fitbit Flex Teardown



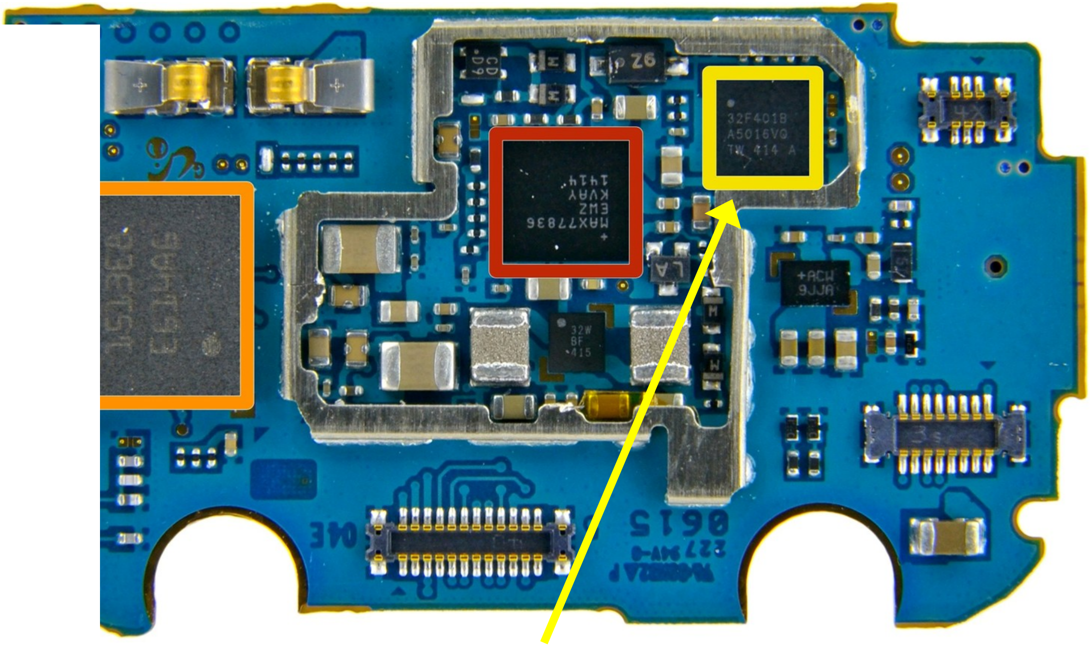
STMicroelectronics 32L151C6 Ultra Low Power ARM Cortex M3 Microcontroller

Nordic Semiconductor nRF8001 Bluetooth Low Energy Connectivity IC

www.ifixit.com

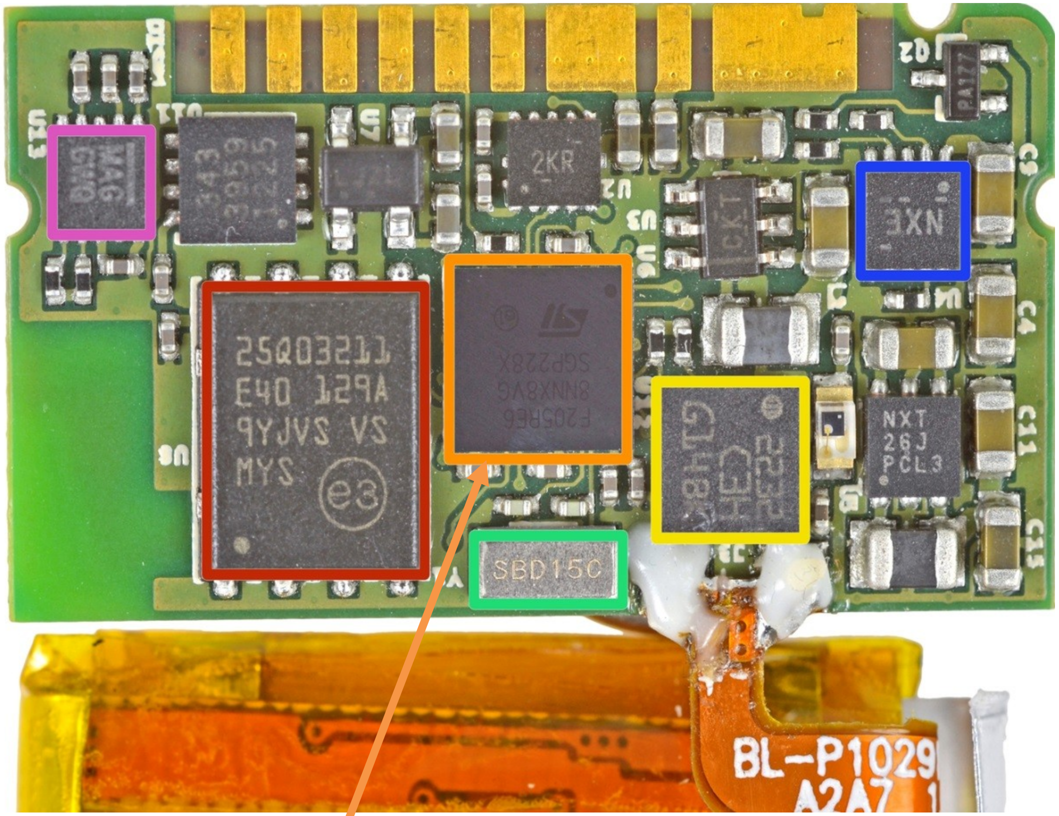
Samsung Galaxy Gear

source: ifixit.com



- STMicroelectronics STM32F401B **ARM-Cortex M4** MCU with 128KB Flash

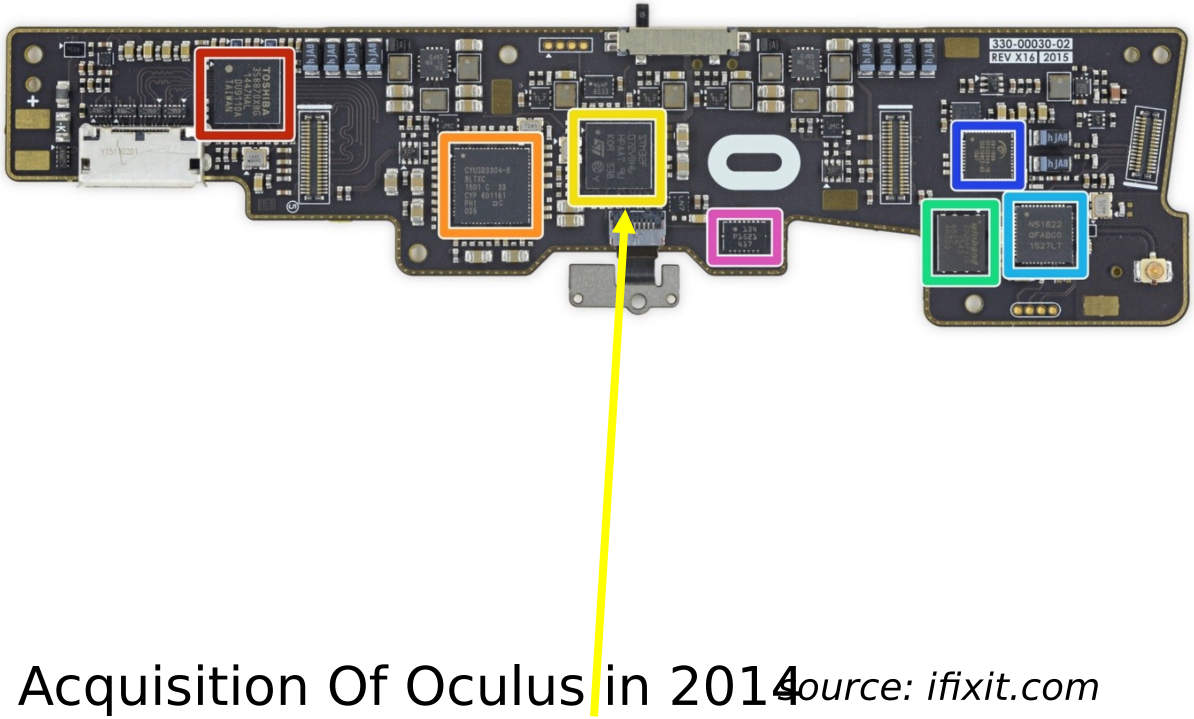
Pebble Smartwatch



source: ifixit.com

- STMicroelectronics STM32F205RE **ARM Cortex-M3** MCU, with a maximum speed of 120 MHz

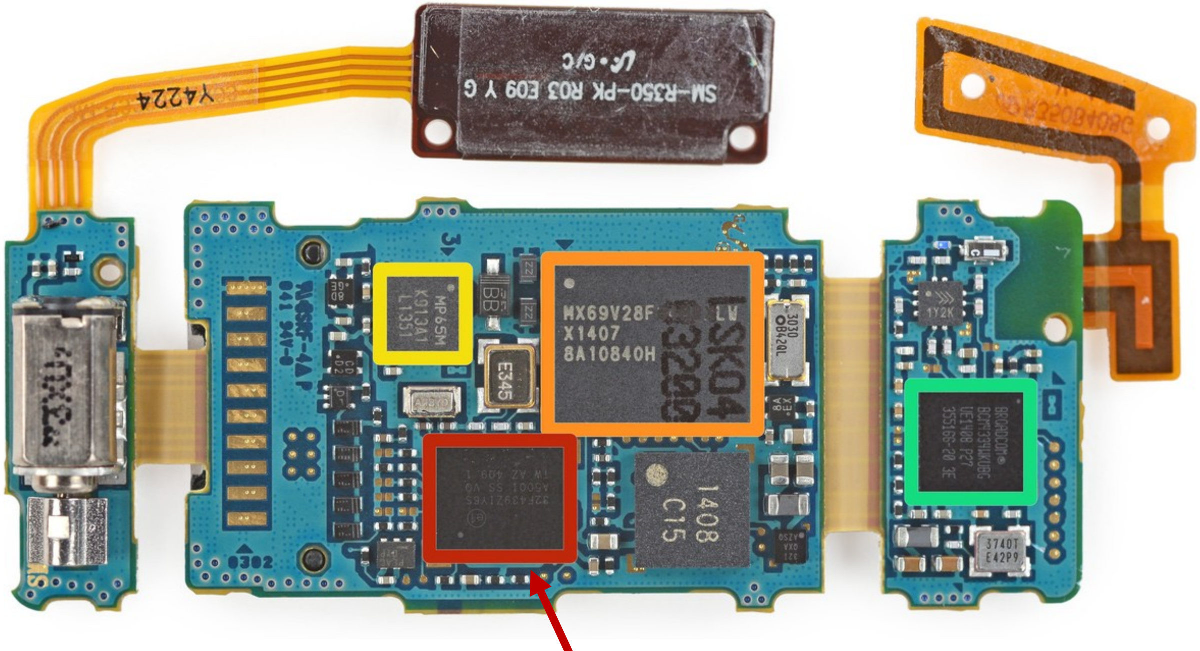
Oculus VR



Source: ifixit.com

- Facebook’s $2 Billion Acquisition Of Oculus in 2014
- ST Microelectronics STM32F072VB **ARM Cortex-M0** 32-bit RISC Core Microcontroller

Samsung Gear Fit Fitness Tracker



source: ifixit.com

- STMicroelectronics **STM32F4**39ZI 180 MHz, 32 bit **ARM Cortex-M4** CPU

Assesment

- Labs 30%
- Project part A – 25%. Technical report on quadcoptor firmware + etc??
- Essay 20%
- Project part B – coding project. - 25%