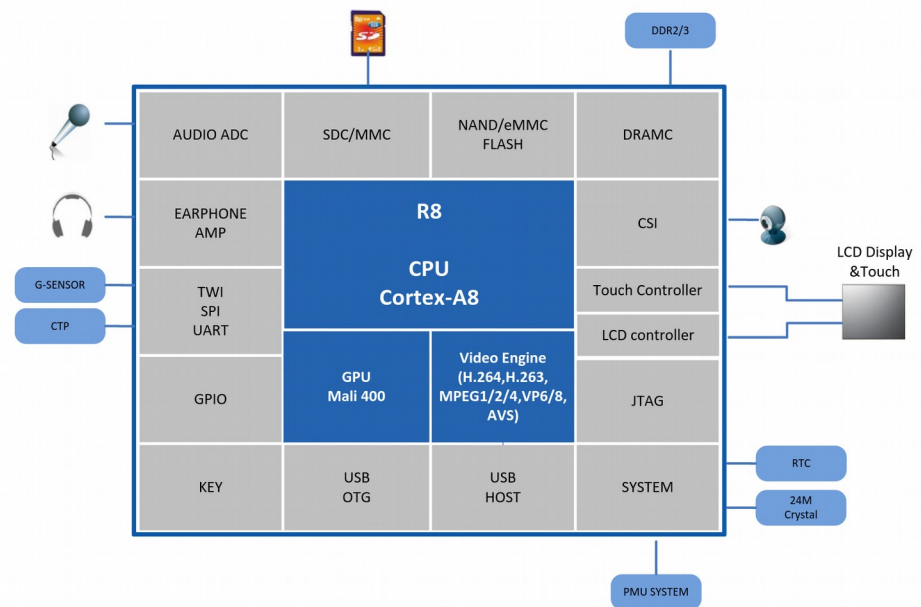


# Embedded Linux

- Syfte: Köra på en (liten) pryl som kan prata med hårdvara
- Hur ser det ut?
  - System On Chip: Processor, grafikkort, minne, ljudkort, kontrollkretsar m.m. i ett chip
  - Vår arkitektur: ARM v7, SoC: Allwinner R8 (se <http://linux-sunxi.org/R8>)



# Embedded Linux

- Hur gör Linux?
  - Linux har en **kärna** som sköter all kommunikation med hårdvaran genom minnet. Kärnan sparas som en avbild (ulmage eller zImage) som går att boota mha en bootloader.
  - Alla drivare är numera utanför kärnan i en **device tree blob**. Syftet är att det ska vara flexibelt och återanvändbar. Hårdvaran specificeras i **device tree source** filer som byggs ihop till en DTB.
- Varför ett träd?
  - Gör det lätt att ärva gemensamma saker, t ex stöd för ARM
- Hårdvarustöd för ett specifikt kort kallas **BSP** (board support package) och ligger i vårt fall under `arch/arm/boot/dts/sun5i-r8-chip.dts` i kärnans källkodsträd

# Embedded Linux

- Hur använder man drivarna?
  - Huvudsakligen finns det två metoder:
    - Minnesmappning (mmap)
    - Genom enhetsfiler (**device files**)
- För GPIO
  - sysfs (se <http://docs.getchip.com/#how-the-system-sees-gpio>)
- Specifikt om I2C:
  - i2cdev
  - Bussarna finns tillgängliga på /dev/i2c-0,1,2
  - Från terminalen genom kommandon (i2cdetect, i2cget, i2cset)
  - Filerna kan manipuleras från C-program genom ioctl (i2c\_smbus\_read\_word\_data, i2c\_smbus\_write\_word\_data)