Real-Time Systems

Week 3

Embedded Systems

Outline

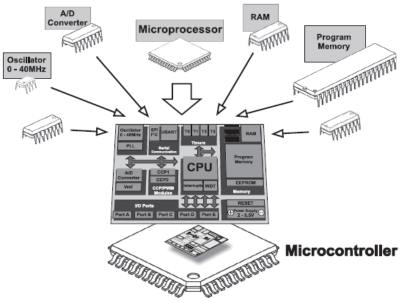
- What is an embedded system
- Microprocessor vs. microcontroller
- Introduction to microcontrollers
- Projects!!

Embedded Systems

- An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.
- They are everywhere!!!
  - House appliances
  - Aircrafts
  - Vehicles
  - so on…

Microprocessor vs. Microcontroller

- To summarize, a microcontroller is a (stripped-down) processor which is equipped with memory, timers, I/O pins and other on-chip peripherals.
- The driving element behind all this is cost; Integrating all elements on one chip saves space and leads to both lower manufacturing costs and shorter development times.
- This saves both time and money, which are key factors in embedded systems.



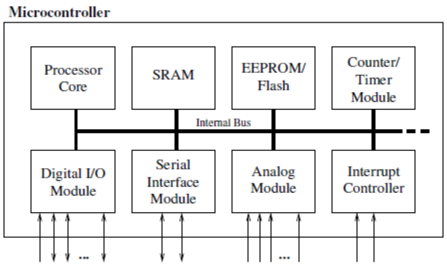
Microprocessor vs. Microcontroller

- Additional advantages of the integration are easy upgradability, lower power consumption, and higher reliability, which are also very important aspects in embedded systems.
- On the downside, using a microcontroller to solve a task in software that could also be solved with a hardware solution will not give you the same speed that the hardware solution could achieve.

Introduction to Microcontrollers

- Components
  - Core, memory, digital I/O, etc.
- Communication interfaces
  - UART, SPI, I2C, etc.
- Programming scheme
  - Howto program, download, debug,
- Peripherals
  - switches, buttons, displays, leds, so on.

Microcontroller Components



- All components are connected via an internal bus and are all integrated on one chip.
- The modules are connected to the outside world via I/O pins.

Microcontroller Components

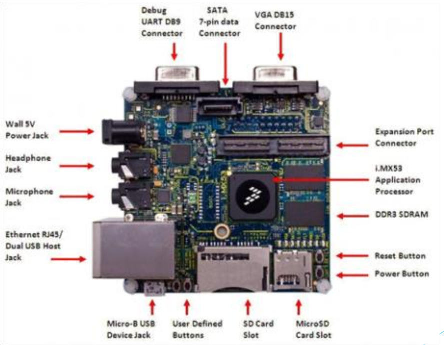
- Processor Core
  - The CPU of the controller. It contains the arithmetic logic unit, the control unit, and the registers (stack pointer, program counter, accumulator register, register file, ...)
- Memory
  - The memory is sometimes split into program memory and data memory. In larger controllers, a DMA controller handles data transfers between peripheral components and the memory.
- Interrupt Controller
  - Interrupts are useful for interrupting the normal program flow in case of (important) external or internal events. In conjunction with sleep modes, they help to conserve power.

Microcontroller Components

- Timer/Counter
  - Most controllers have at least one and more likely 2-3 Timer/Counters, which can be used to timestamp events, measure intervals, or count events.
  - Many controllers also contain PWM (pulse width modulation) outputs, which can be used to drive motors or for safe breaking (antilock brake system, ABS). Furthermore the PWM output can, in conjunction with an external filter, be used to realize a cheap digital/analog converter.
- Digital I/O
  - Parallel digital I/O ports are one of the main features of microcontrollers. The number of I/O pins varies from 3-4 to over 100, depending on the controller family and the controller type.

Microcontroller Components

- Analog I/O
  - Apart from a few small controllers, most microcontrollers have integrated analog/digital converters, which differ in the number of channels (2-16) and their resolution (8-12 bits).
  - The analog module also generally features an analog comparator.
  - In most cases, the microcontroller includes digital/analog converters.



Microcontroller Components

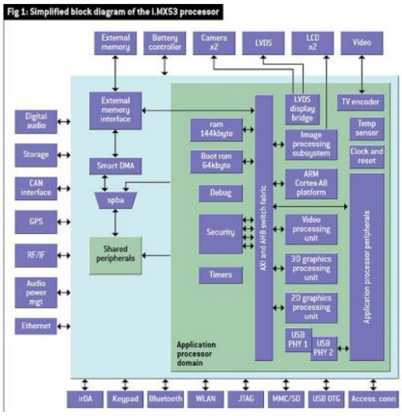


Fig 1: Simplified block diagram of the i.MX53 processor

- Interfaces
  - Controllers generally have at least one serial interface which can be used to download the program and for communication with the development PC in general.
  - Since serial interfaces can also be used to communicate with external peripheral devices, most controllers offer several and varied interfaces like ***SPI and SCI***.
  - Many microcontrollers also contain integrated bus controllers for the most common (field)busses. ***I2C and CAN*** controllers lead the field here. Larger microcontrollers may also contain ***PCI, USB, or Ethernet*** interfaces.

Microcontroller Components

- Watchdog Timer
  - Since safety-critical systems form a major application area of microcontrollers, it is important to guard against errors in the program and/or the hardware. The watchdog timer is used to reset the controller in case of software “crashes”.
- Debugging Unit
  - Some controllers are equipped with additional hardware to allow remote debugging of the chip from the PC. So there is no need to download special debugging software, which has the distinct advantage that erroneous application code cannot overwrite the debugger

Programming on Microcontrollers

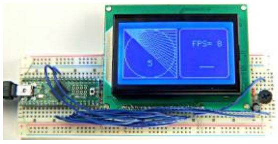
- Bare machine / Bare metal
  - system boots directly into monolithic, single-purpose software, without loading a separate operating system
  - usually written in assembly and/or C
- Application on a operating system
  - Industrial PC, Arm based-microcontrollers, etc.
  - Linux, Windows Embedded

Programming on Microcontrollers

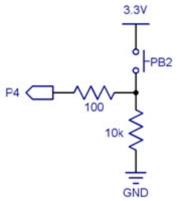
- Programming is done mostly on a PC
  - a special editor / IDE
- Cross compilation is required.
  - building an executable for a different architecture
- Transfer binary code to microcontroller
  - A programmer is required.
- Debugging
  - no printf!!
  - peripherals (leds, communication channels)
  - remote debugging

Peripherals

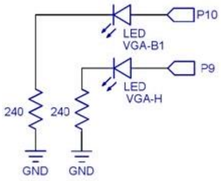
- Displays
  - 7 segment
  - dot matrix lcd



- Buttons



- Leds



Projects

- What we have…
  - Almost everything ☺
- Development boards for several microcontrollers
  - 8-bit/16-bit/32-bit
  - Atmel atmega
  - Texas Instruments MSP series
  - ARM based board; Friendly arm, i.Mx53 boards
  - Ardunio

Projects

- Peripherals
  - leds, buttons, displays,
  - rf boards,
  - seismic sensor
  - cameras
  - so on…

Projects

- Sample projects
  - iPot
    - a smart pot which is able to monitor humidity, heat and conductivity of the soil.
  - Eartquake detector
    - a system measuring the amplitude of seismic waves and sounding an alarm.