1st Summer School on ADVANCED TECHNOLOGIES FOR NEURO-MOTOR ASSESSMENT AND REHABILITATION

Design of Sensor Based Mobile Platforms

Dr. Luca Roffia

ARCES – University of Bologna

Contacts:
lroffia@arces.unibo.it
+39 051 2095421/6

18 – 24 June 2006, Monte S. Pietro - **BOLOGNA, ITALY**

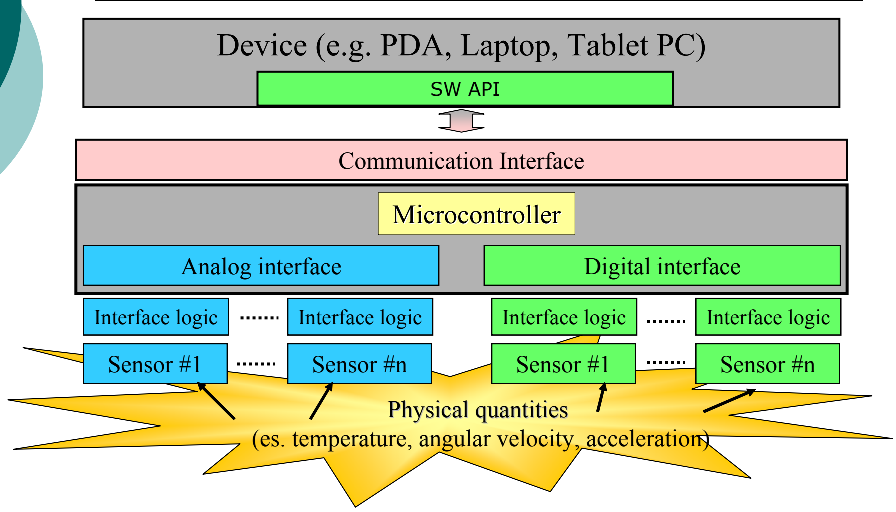
Overview

- This seminar discusses the main software issues involved in the integration of sensors in mobile and possibly "wearable" platforms.
- Sensors can be used both in the interest of the user and to support platform resource management (e.g. for power optimization purposes).
- For this reason sensors will become in the future a shared system level resource much in the same way as the network interface is today; but sensor management facilities are not yet available in current platforms therefore sensor handling needs still to be treated as an application level issue.
- The seminar first briefly reviews a research approach to system level sensor management in ubiquitous systems, and afterwards concentrates on current sensors integration issues, focusing on the following relevant aspects:
  - Acquisition and processing of data originating from digital and analog sensors (accelerometers, gyroscopes and compasses);
  - Sensors calibration procedures and common noise sources;
  - Fixed point arithmetic;
  - Sensors remotization for improved wearability with Simple Wireless Interfaces (ZigBee and BlueTooth)

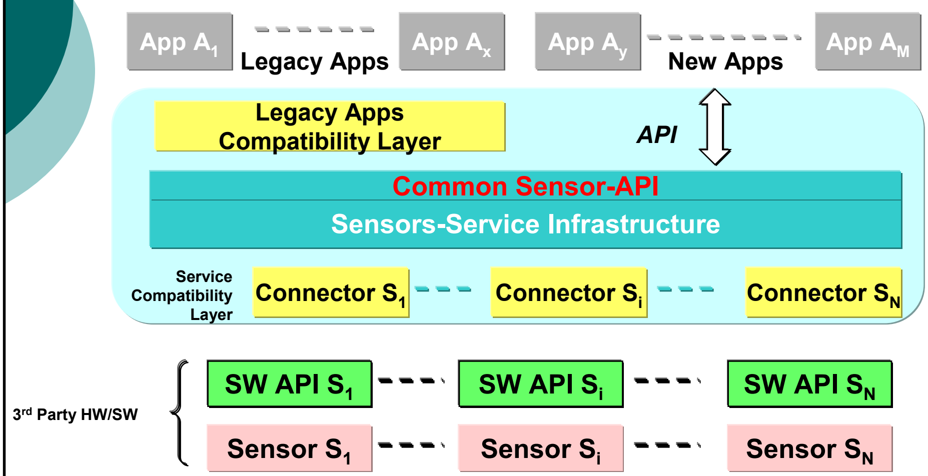
Summary

- Embedded sensor platform
- Sensors as system level resources
- Demo
  - Tracking and remote content control
- Methodological approach, common error sources and constraints
- Case of study: a heading system
  - Examples of acquisition and calibration of different analog and digital sensors
    - Accelerometer
    - Gyroscope
    - Compass
- Fixed point arithmetic

Embedded Sensor Platform

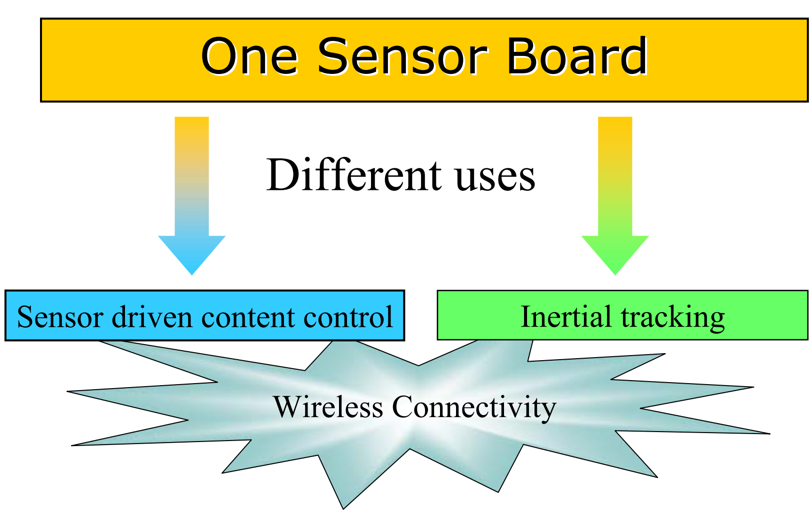


Sensors as system resources



INTEL Internship, courtesy of Giuseppe Raffa

Sensor Mobile Platform DEMO



Methodological approach

- **What we need sensors for?**
  - E.g. Heading system, tracking system, VR system, human activities analysis
- **Which sensors can be used for our purposes?**
  - E.g. accelerometer, compass, light, pressure
  - The final output of the sensor system is not merely related to the physical quantities sensed
- **Sensor signal analysis**
  - Data sheets are not enough
  - Sensor and circuit parameters differ from system to system
  - E.g. Dynamic range, noise, bandwidth
- **Calibration procedures**
  - Find the right compromise between precision and be easy to use and to implement (one time calibration)
  - Often you need an ad-hoc mechanical setup
- **Software-hardware co-design**
  - Software means flexibility
  - Depending on team experiences on analog or digital processing
  - Signal conditioning: always an analog process
  - Signal processing: hardware, software, mixed
  - Find the right balance

Common error sources

- Noise on supply voltage
  - Hardware filters
  - Ad-hoc power sections
- Temperature
  - Calibration
  - Automatic compensation
- Sensors cross references
  - Gyroscope and compass ➔ tilt
  - Accelerometer ➔ rate
- Environmental disturbances
  - Magnetic field distortion (e.g. Iron)
- Software
  - ADC
  - Synchronization and timing
  - Fixed point arithmetic

Common constraints

Power consumption
Size and weight
Low cost
Wireless Connectivity
Memory constraints

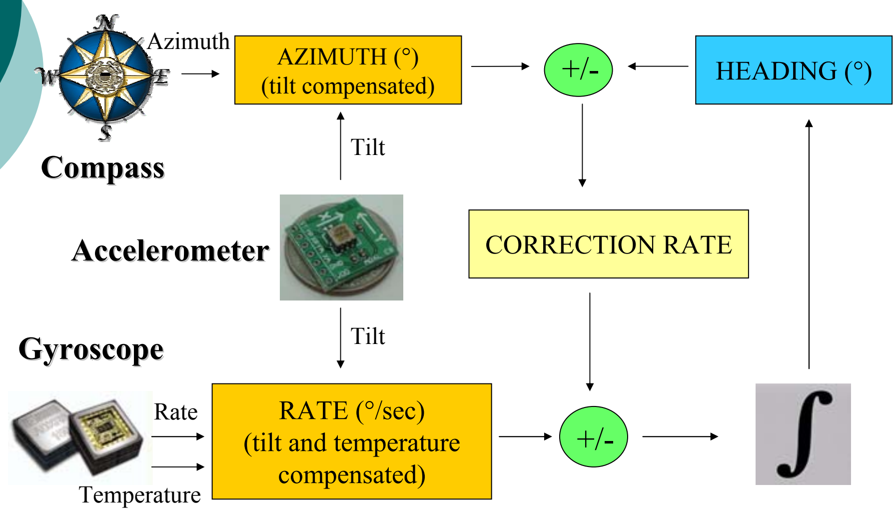
More power needed ➔ larger and heavier implementation
Low power components ➔ more expensive solution

TRADE OFF
to be searched based on ergonomics requirements

Case study: a Heading System

- Accelerometer:
  - Analog Devices ADXL202 (Digital)
- Gyroscope:
  - Analog Devices ADXR150 (Analog)
- Compass:
  - Based on Philips KMZ51(Analog)
- Microcontroller:
  - Texas Instrument MSP430F149

Heading system: conceptual schema



Microcontroller characteristics

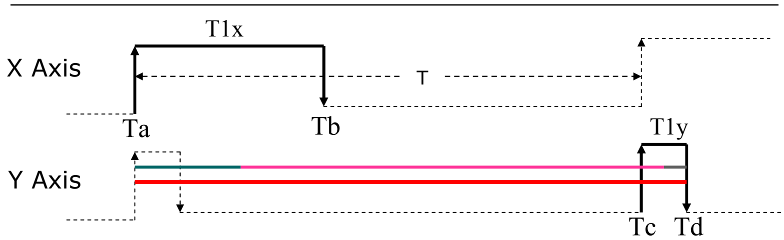
- Low Supply-Voltage Range, 1.8 V . . . 3.6 V
- Ultralow-Power Consumption:
  - Active Mode: 280 μA at 1 MHz, 2.2V
  - Standby Mode: 1.6 μA
  - Off Mode (RAM Retention): 0.1 μA
- Five Power-Saving Modes
- Wake-Up From Standby Mode in less than 6 μs
- 16-Bit RISC Architecture, 125-ns Instruction Cycle Time
- 12-Bit A/D Converter With Internal Reference, Sample-and-Hold and Autoscan Feature
- 16-Bit Timer_B With Seven Capture/Compare-With-Shadow Registers
- 16-Bit Timer_A With Three Capture/Compare Registers
- On-Chip Comparator
- Serial Onboard Programming, No External Programming Voltage Needed Programmable Code Protection by Security

- Serial Communication Interface (USART), Functions as Asynchronous UART or Synchronous SPI Interface
  - Two USARTs (USART0, USART1) — MSP430x14x(1) Devices
  - One USART (USART0) — MSP430x13x Devices
- Family Members Include:
  - MSP430F133: 8KB+256B Flash Memory, 256B RAM
  - MSP430F135: 16KB+256B Flash Memory, 512B RAM
  - MSP430F147, MSP430F1471†: 32KB+256B Flash Memory, 1KB RAM
  - MSP430F148, MSP430F1481†: 48KB+256B Flash Memory, 2KB RAM
  - MSP430F149, MSP430F1491†: 60KB+256B Flash Memory, 2KB RAM

Accelerometer

- Range: ±2g
- Digital:
  - PWM: 50 % ➔ 0 g ; Δ 12,5 % ➔1g
- Filtering:
  - Hardware 15 Hz for human activities recognition and motion detection
  - Software: Low-pass 1 Hz filtering for continuous component (tilt detection)
- One time calibration:
  - Offset: still on a horizontal plane
  - Scale factor: one known angle for each axis (e.g. 90°)
    - Note: to enhance the precision within a specific range of use select a more suitable calibration angle (e.g. 45°)
- Timers
  - Capture (raising edge, falling edge)

Accelerometer PWM encoding algorithm



$$T2 = (Td - Ta) - (Tb - Ta)/2 - (Td - Tc)/2$$

$$T1x = (Tb - Ta)$$

$$T1y = (Td - Tc)$$

T2 = f(Temperature) ➔ Automatic temperature compensation

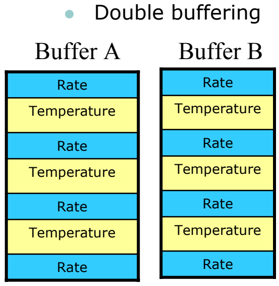
Acceleration calculation

- OFFSET [%]
  - 0g ➔ $T1x0$, $T1y0$, $T20$
    - PWMX@0g = T1x0 / T20
    - PWMY@0g =T1y0 / T20
- SCALE FACTOR [(m/sec^2)/%]
  - PITCH (ρ°) ➔ AccX@ρ = g * sen(ρ)
  - ROLL (Φ°) ➔ AccY@Φ = g * sen(Φ)
    - Sfx = AccX@ρ / (PWMX@ρ - PWMX@0g)
    - Sfy = AccY@Φ / (PWMY@Φ - PWMY@0g)
- ACCELERATION [m/sec^2]
  - Xg ➔ T1x, T1y, T2
    - Δ PWMX = (T1x / T2) - PWMX@0g
    - Δ PWMY = (T1y / T2) - PWMY@0g

ACCX = Sfx * Δ PWMX
ACCY = Sfy * Δ PWMY

Gyroscope

- Range: ±150 °/sec
- Analog sensor: dynamic range 0 – 5 V
- Nominal offset and scale factor: 1,65 V , 12.5 mV/(°/sec)
- ADC
  - Rate and temperature together (easier to be implemented but not so efficient) ➔ temperature varies very slowly
  - Double buffering

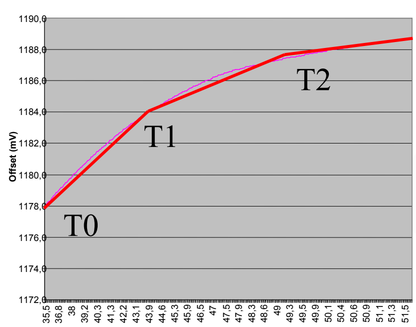


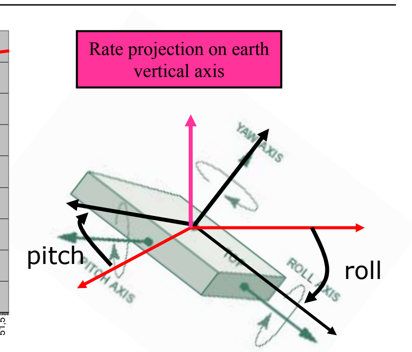
Each time a buffer is full an interrupt is generated and the microprocessor processes the sampled data. Meanwhile the ADC continues to sample in the other buffer

Gyroscope calibration

- **Offset**
  - still on a horizontal plane @ T0(**Voffset**)
- **Scale factor**
  - A 360° rotation on a horizontal plane (clockwise, anti-clockwise) using high precision rotating table (**Sf_h**)
- **Temperature compensation**
  - Voffset_temp = F(temperature, Voffset)
- **Tilt compensation**
  - Static component of acceleration (pitch, roll)
  - Accelerometer as close as possible to the rotation axis

Gyroscope temperature and tilt compensation



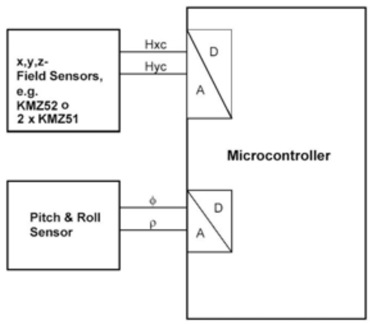


Voffset_temp = F(temperature, **Voffset**)

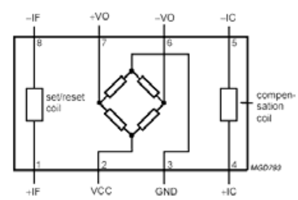
Sf_comp = cos(pitch)*cos(roll)* **Sf_h**

Rate = (Vadc – Voffset_temp) / Sf_comp

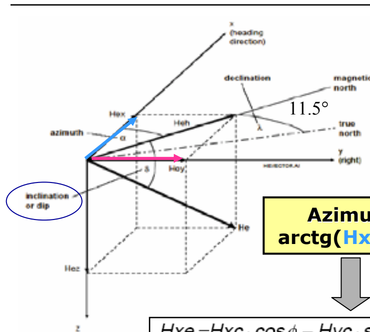
Tilt compensed compass

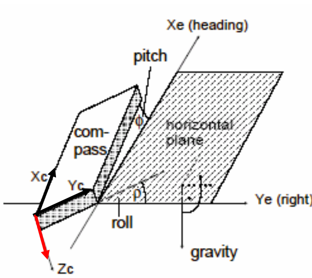


- Magneto resistive effect
  - *Resistance changes depending on the magnetic field*
- Flipping current (±1A for at least 3usec)
- As consequence the differential voltage (+V0, -V0) changes



Effect of tilt on azimuth





Azimuth = arctg(Hxe/Hye)

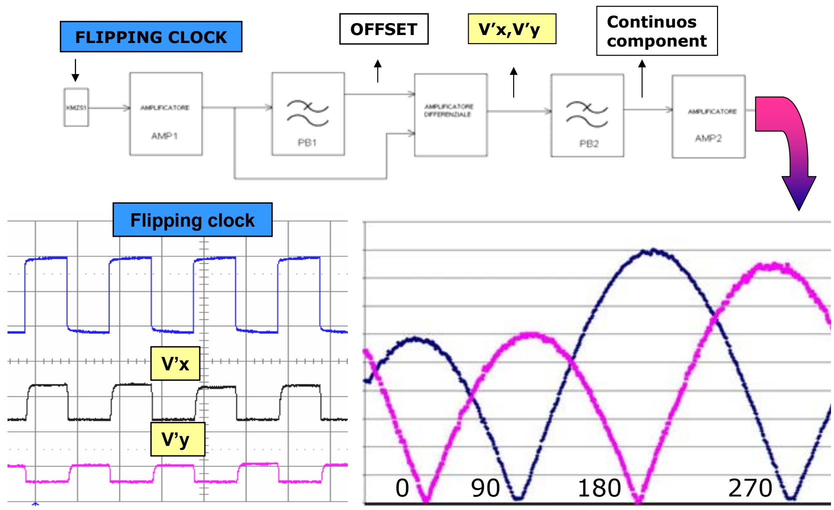
$$Hxe = Hxc \cdot \cos\phi - Hyc \cdot \sin\phi \cdot \sin\rho - Hzc \cdot \sin\phi \cdot \cos\rho$$
$$Hye = Hyc \cdot \cos\rho + Hzc \cdot \sin\rho$$

5 degrees of freedom:
- **Φ** (*pitch*) , **ρ** (*roll*)
- **Hxc**, **Hyc**, **Hzc** = magnetic field respectively on compass X,Y,Z axis

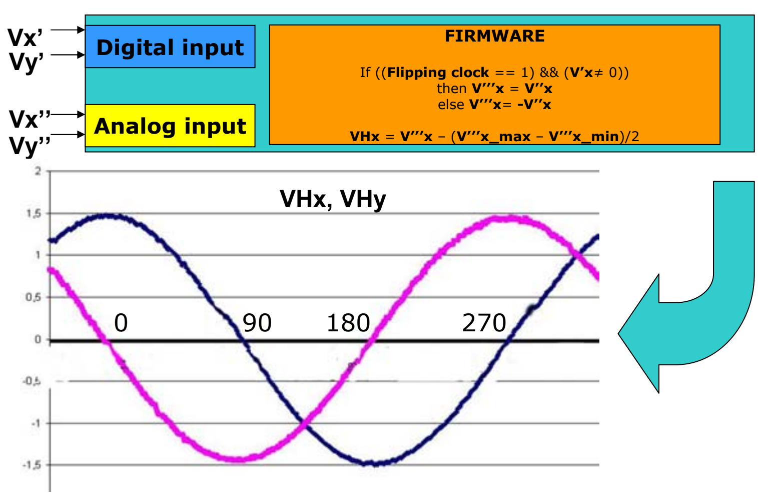
Hzc = F (**Hxc, Hyc, Φ, ρ,** dip)

dip = F(**latitude**)

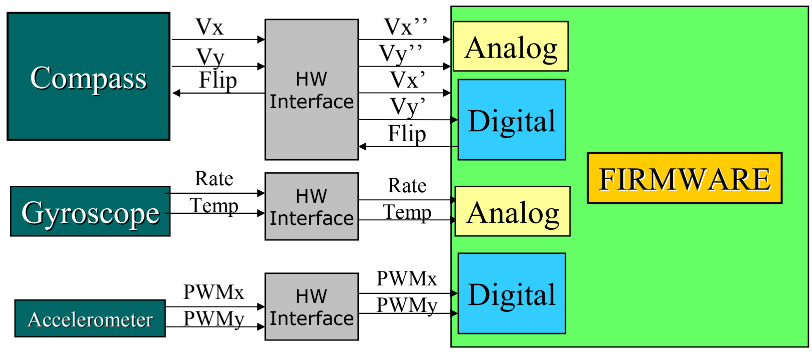
Analog processing chain



Digital processing chain



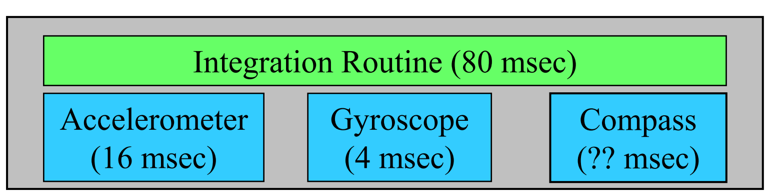
Heading Sensor Board Architecture



Power consumption: **80 mA** in continuos mode (no power optimizations)
Low cost : **80 €**
Wireless connection: **BlueTooth** and **ZigBee** Interfaces
Dimension and weight: mainly depend on batteries

Synchronization

- Timers: Compare mode ➔ generate time events



Additional processing resources are needed for communication

Not only the best timing to acquire a single signal but also a **balance of the entire system timing** should be considered

Fixed Point Arithmetic

- **Rounding** : integer division
  - A/B ➔ (A + B/2) /B
  - es. A=9 B=5
  - A/B = 1,8 = (int) 1 !!
  - (A + B/2) /B = (9 + 2)/5 = (int) 2 OK
- **Scaling**
  - N,D1D2D3 ➔ N D1D2D3 (scale factor = 1000)
  - Es. Scale factor = 10
  - Floating point : 7 / 8 = 0,875 = 0,9
  - Fixed point: 7,0 / 8 ➔ 70 / 8 = 8,75 = (int) 8 !! ➔ rounding ➔
  - (70 + 4) / 8 = 9,25 = (int) 9 OK
- **Data range**
  - Use the right primitive type (don't waste memory)
  - On a 16-bit microprocessor:
  - Char ➔ 8 bits
  - Int ➔ 16 bits
  - Long ➔ 32 bits
  - Very important to distinguish between signed and unsigned

Bibliography

- **ADXL202 Data Sheet**, http://www.analog.com/en/prod/0,2877,ADXL202,00.html
- **KMZ51 Data Sheet**, http://www.semiconductors.philips.com/pip/KMZ51.html
- **ADXRS150 Data Sheet**, http://www.analog.com/en/prod/0,2877,ADXRS150,00.html
- **MSP430F149 Data Sheet,** http://focus.ti.com/docs/prod/folders/print/msp430f149.html
- **A Calibration Technique for a Two-Axis Magnetic Compass in Telematics Devices**, Seong Yun Cho and Chan Gook Park ,ETRI Journal, Volume 27, Number 3, June 2005 http://journal.etri.re.kr/Cyber/servlet/BrowseAbstract?paperid=RP0412-0192
- **Using the ADXL202 in Pedometer and Personal Navigation Applications by Harvey Weinberg, Analog Device AN-602,** http://www.analog.com/UploadedFiles/Application_Notes/513772624AN602.pdf
- **Using the ADXL202 Duty Cycle Output by Harvey Weinberg, Analog Device AN-602 ,** http://www.analog.com/UploadedFiles/Application_Notes/320058905AN604.pdf
- **A Calculated Look at Fixed-Point Arithmetic** , Robert Gordon , Programming Embedded Systems, http://www.embedded.com/98/9804fe2.htm
- **Embedded Systems Design** http://www.embedded.com/