INTRODUCTION TO
ESIPAP COMPUTING SESSIONS

WEDNESDAY 13 – THURSDAY 14 FEBRUARY 2019
GOALS OF THE COMPUTING SESSIONS

• Computing is required for instrumentation purposes:
  • Simulation of sensor
  • Data acquisition
  • Data analysis
  • Algorithm and reconstruction of physics objects

• Computing sessions target to apply your theoretical knowledge:
  • Instrumentation
  • Software programming in C++
  • Using specific tools of high energy physics: ROOT and Geant4

• Working by yourself and experimenting

• Getting the good practice
PHYSICS CONTEXT
THE CMS (COMPACT MUON SOLENOID) DETECTOR
THE CMS (COMPACT MUON SOLENOID) DETECTOR
SILICON STRIP TRACKER
CMS silicon strip tracker in few numbers:
- 15,000 modules
- Surface: ~ 200 m²
- $10^6$ channels

Instrumental activities
- R&D
- Construction
- Operation (online)
- Alignment & calibration
- Offline analyses
- Simulation
- Radiation damages evaluation
- ...

Performances:
- Hit resolution: 20-40 μm
- Hit efficiency > 98% (at high Pile-Up)
- Timing alignment accuracy: 1ns
- ...
During its operation it is important to monitor environment conditions:

- Temperature
- Leakage current
- Noise
- Thermal dissipation
- Radiation damages
- …
- Humidity
- Dew points & condensation
- Front End electronics
- …

Monitoring tools
Several probes are used to monitor that:

- On-board sensors
- External sensors
→ Some are ARDUINO-based!
1. **Slow control**
   - Using a dedicated electronic board (Sense Hat) read by a Raspberry
     - Monitor the temperature & humidity
     - Send warning when conditions are not fulfilled

2. **Offline analyses**
   - Calibration of the temperature sensors
   - Evaluation of the sensor resolution

3. **Simulation**
   - Basic simulation with the GEANT4 package of a CMS silicon strip sensor

**Instrumental activities**

- R&D
- Construction
- **Operation (online)**
- Alignment & calibration
- **Offline analyses**
- **Simulation**
- Radiation damages evaluation
- ...

**COMPUTING SESSION AIMS**
THE RASPBERRY BOARD

Raspberry Pi 3 B+ motherboard

- Quad-core 64 bits processors @ 1.4 GHz
- ARM (Acorn Risc Machine) architecture used mainly in smartphones, tablets, robotics, automation

Advantages: price, flexibility, performances
CONNECTIONS TO PERIPHERICAL DEVICES

- 4 USB ports
- 1 ethernet port
- 1 HDMI plug
- 1 GPIO (General Purpose Input/Output) port for connecting sensors
- Powered by micro USB (5V, 2.5A min)

+ WIFI
+ Bluetooth
LINUX DISTRIBUTION: RASPBIAN

Stored on a micro SD card
SENSE HAT BOARD

- 8x8 LEDs for display
- Pressure / Temperature sensor
- Joystick
- Humidity / Temperature sensor
- 3D accelerometer, 3D gyrometer and 3D magnetometer sensor
PRICE

Raspberry Pi 3 B+  
~ 40 €

Sense Hat  
~ 30 €

Connectors  
~ 15 €

Micro SD  
~ 10 €

Total: ~ 100€  
(good gift for Saint-Valentin’s day)
SENSORS TO STUDY

- Humidity / Temperature sensor
- Pressure / Temperature sensor
HOW TO MEASURE ....?

Temperature

\[ R = \rho \frac{S}{L} \]

Material resistivity \( \rho \) depends on temperature.

Pressure

Piezoresistive effect:

a change in resistivity when a stress is applied.

Humidity

Dielectric material absorbs water molecules until equilibrium

\[ \rightarrow \] change the electrical conductivity \( \varepsilon \) [in S/m]
PRESSURE / TEMPERATURE SENSOR PROCESS
Wheatstone bridge for translating change of resistance into change of tension
Analogic Front-End

- Small signal voltages vs noise floor
- Amplifying signal and removing noise
PRESSURE / TEMPERATURE SENSOR PROCESS

Multiplexer
Treating pressure and temperature measures by the same channel
PRESSURE / TEMPERATURE SENSOR PROCESS

Analogic to Digital converter
Digitalization of the measure
Temperature compensation
Piezoresistivity depends on T
→ Need to compensate this effect

Calibration settings
A maximum of 32 successive measurements are done (~1s) and an average value is computed.
Data transfer
Data are sent to the Raspberry via the GPIO port with the protocol I2C
One logic part is missing in this schema:

Translation of the tension to temperature and relative humidity.
ADC RESOLUTION FOR TEMPERATURE SENSOR

- Operating range of the sensor: [260 hPa to 1260 hPa] where the sensor is relevant and reliable
- Conversion pressure to measure: \( \text{measure} = \text{pressure} \times 4096 \)
- Number of bits for coding the maximum value 1260 hPa → \( \text{measure} = 5 \times 10^6 \) → \( N = 23 \) bits because \( 2^{22} < \text{measure} < 2^{23} - 1 \) .... but not standard: using 24 bits – ADC
- Full range: [0 hPa to 4096 hPa]
- Sensitivity: \( 4096 \text{ hPa} / 2^{24} - 1 = 0.00024 \text{ hPa} \)
# SUMMARY ON ADC SENSITIVITY

<table>
<thead>
<tr>
<th></th>
<th>Pressure - Temperature sensor</th>
<th>Humidity - Temperature sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pressure</strong></td>
<td>260 hPa to 1260 hPa</td>
<td>0% to 100%</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>-30°C to +105°C</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Humidity</strong></td>
<td>0% to 100%</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>0% to 100%</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Pressure</strong></td>
<td>260 hPa to 1260 hPa</td>
<td>0% to 100%</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>-30°C to +105°C</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Humidity</strong></td>
<td>0% to 100%</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>0% to 100%</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Operating range</strong></td>
<td>260 hPa to 1260 hPa</td>
<td>0% to 100%</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>-30°C to +105°C</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Humidity</strong></td>
<td>0% to 100%</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>0% to 100%</td>
<td>-40°C to +120°C</td>
</tr>
<tr>
<td><strong>Full scale</strong></td>
<td>0 hPa to 4096 hPa</td>
<td>-30°C to +110°C</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>-30°C to +110°C</td>
<td>Linear interpolation,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>depending of the calibration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>coefficients</td>
</tr>
<tr>
<td><strong>ADC resolution</strong></td>
<td>24 bits</td>
<td>16 bits</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>16 bits</td>
<td>16 bits</td>
</tr>
<tr>
<td><strong>Sensitivity</strong></td>
<td>0.00024 hPa</td>
<td>0.002 °C</td>
</tr>
<tr>
<td></td>
<td>0.004 %</td>
<td>0.016 °C</td>
</tr>
<tr>
<td></td>
<td>0.004 %</td>
<td>0.016 °C</td>
</tr>
</tbody>
</table>
ORGANIZATION
ORGANIZATION IN SESSIONS

**Wednesday**
- **Session 1**
  - Introduction
  - Data acquisition

**Thursday**
- **Session 4**
  - Analyzing data with ROOT
- **Session 5**
  - Simulating particle interaction with GEANT4
  - Summary

**Sessions**
- Session 1
- Session 2
- Session 3
- Session 4
- Session 5

**Schedule**
- 9:00
- 12:15
- 14:00
- 17:15
- 17:20
- 18:50
# ONE STUDENT, ONE RASPBERRY, ONE PC

<table>
<thead>
<tr>
<th>First Name</th>
<th>Family Name</th>
<th>Email address</th>
<th>Raspberry ID</th>
<th>PC ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Google Spreadsheet](https://docs.google.com/spreadsheets/d/11kUn_jwpypZLoiuCy6KbL9pOtJx85zv2GlgtNlCG9CQ/edit?usp=sharing)
SAVING YOUR PRODUCTION

- End of session 2
- End of session 4

Sending your code to the supervisors for assessment

- Using a web service: [www.wetransfer.com](http://www.wetransfer.com)
  - destination: eric.conte@iphc.cnrs.fr
  - author: filling your address email

- A URL link is created: put it on the following spreadsheet

[https://docs.google.com/spreadsheets/d/1QF1hbePGmzLfQ4jQGMD180lx8dLivxRkPI5sIzGXBO8/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1QF1hbePGmzLfQ4jQGMD180lx8dLivxRkPI5sIzGXBO8/edit?usp=sharing)
SKILL ASSESSMENT

### Computing sessions 2019: assessment skill list

<table>
<thead>
<tr>
<th>Skill category</th>
<th>Minimum</th>
<th>Satisfying</th>
<th>Very satisfying</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Knowing C-programming basics</td>
<td>• Writing a “Hello World!” program</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asking questions to the user</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Writing functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Using the standard library</td>
<td>• Using std::cout, std::string, std::stream</td>
<td>• Using std::vector, std::stringstream and cmath.</td>
<td>• Using algorithms, iterators and manipulators.</td>
</tr>
<tr>
<td>3. Writing a C++ class</td>
<td>• Writing a simple class with constructor without and with arguments, destructor, mutators, accessors and “print” function, instantiating and testing the implemented class.</td>
<td>• The class is operational; it fills up the primary aim.</td>
<td>• Implementing operator overloading and copy constructor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Using properly the reserved keyword “const”.</td>
</tr>
<tr>
<td>4. Coding algorithms</td>
<td>• Algorithms work and give the correct results.</td>
<td>• The code is robust: it is protected against bad inputs.</td>
<td>• The code is efficient: efforts are achieved for saving time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Managing properly the dynamic memory allocation (delete).</td>
<td></td>
</tr>
</tbody>
</table>

- Evaluation over 8 categories
- For validating the module
  - Minimum level must be reached for all the 8 categories
  - Satisfying level for at least 3 categories