Οι ανιχνευτες σωματιδιων του LHC ειναι απο τις εντυπωσιακοτερες μηχανες...

...και ισως προσφερονται σαν πιθανες ιδεες για μετρησεις με τους μαθητες σας

(https://www.mycloud.ch/s/S00CA3ADC5EB87832A4A2D2BF2BFB58126F8B438C91)

Α. Τσιρου (ΕΚΠΑ), Piero Giorgio Verdini (INFN Pisa)

Το CERN ειναι το μερος που «παιρνουν σαρκα και οστα» τρια πραγματα:

1. Γινεται βασικη ερευνα για να επιβεβαιωσει η απορριψει τα μοντελα που περιγραφουν το συμπαν και την εξελιξη του. Η ερευνα αυτη απαιτει την δημιουργια, συντηρηση, και λειτουργια οργανων που μπορουν να δωσουν δεδομενα σε στατιστικα ικανες ποσοτητες για τις παραπανω μελετες.
2. Η δημιουργια/εξελιξη και χρηση τεχνολογιας ειναι δεδομενη και η εκφραση «τεχνολογια αιχμης » ειναι σχεδον κενη διοτι το προσωπικο του CERN δημιουργει διαρκως τεχνολογια για να κανει δυνατη την πραγματοποιηση του πρωτου στοχου. Η τεχνολογια αυτη εχει ξεκιναει απο το πραγματικο βαρυ hardware (τροποι χτησιματος και στηριξης) μεχρι το πιο αφαιρετικο software (ποιος δεν ξερει το WEB!).
3. Οι ανθρωποι δουλευουν πραγματικα μαζι και αποδεικνυουν, με συγχωρειτε για την παραφραση το:
   «Εξ ανθρώπου τα χείρω...Και εξ ανθρώπου τα κρείττω»

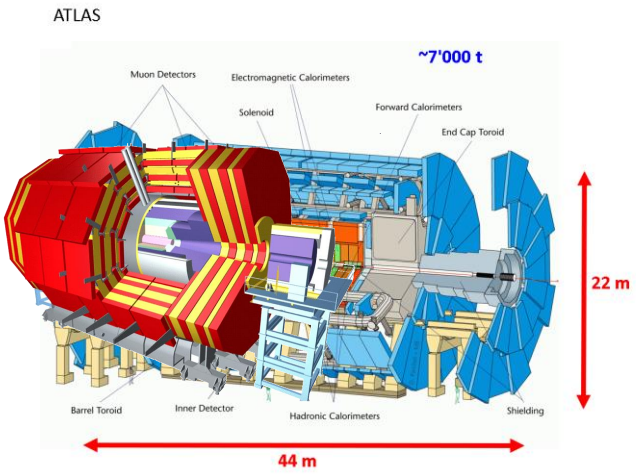
Ετσι, ειναι το μερος που οι νεοι ανθρωποι λατρευουν να δουλευουν ΜΑΖΙ, και ειναι το μερος που λατρευει τους νεους ανθρωπους.

Γι' αυτό εισαστε και ειμαστε εδω.....

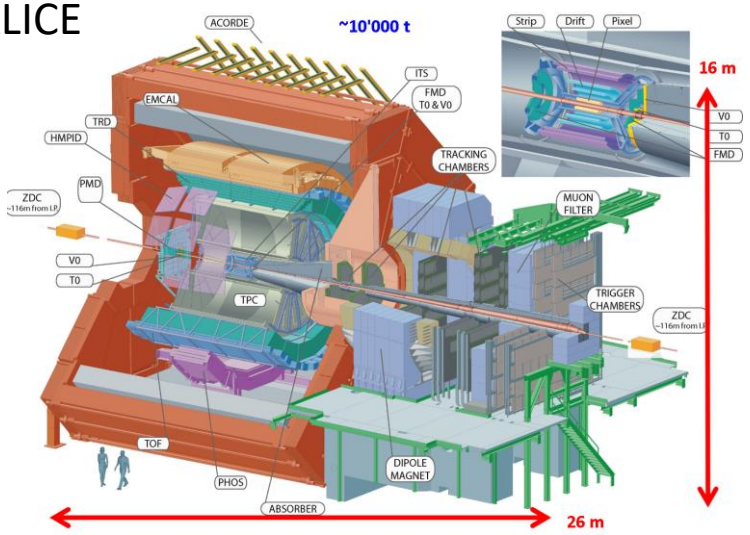
Τι κανουν οι ανιχνευτες:

1. Εξαναγκαζουν τα φορτισμενα σωματιδια που εχουν ενεργεια οσο τα φωτονια πρωτονια και αλλα βασικα σωματιδια τα πρωτα δευτερολεπτα απο την αρχη του Big Bang να μεταμορφωθουν περασουν απο διαφορα ενεργειακα σταδια οπως θεωρητικα εγινε και με το Συμπαν.
2. Καταγραφουν η φωτογραφιζουν αυτα τα σωματιδια και τις συνθηκες κατω απο τις οποιες δημιουργηθηκαν.
3. Ειναι κατασκευασμενοι και λειτουργουν ουτως ωστε να μπορουν να προσφερουν αρκετα δεδομενα ωστε να δωσουν αποτελεσματα με στατιστικη βεβαιοτητα.

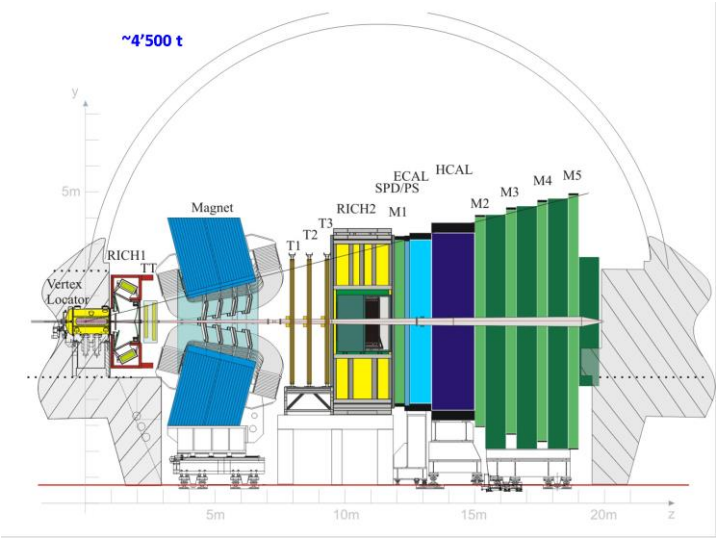
ATLAS



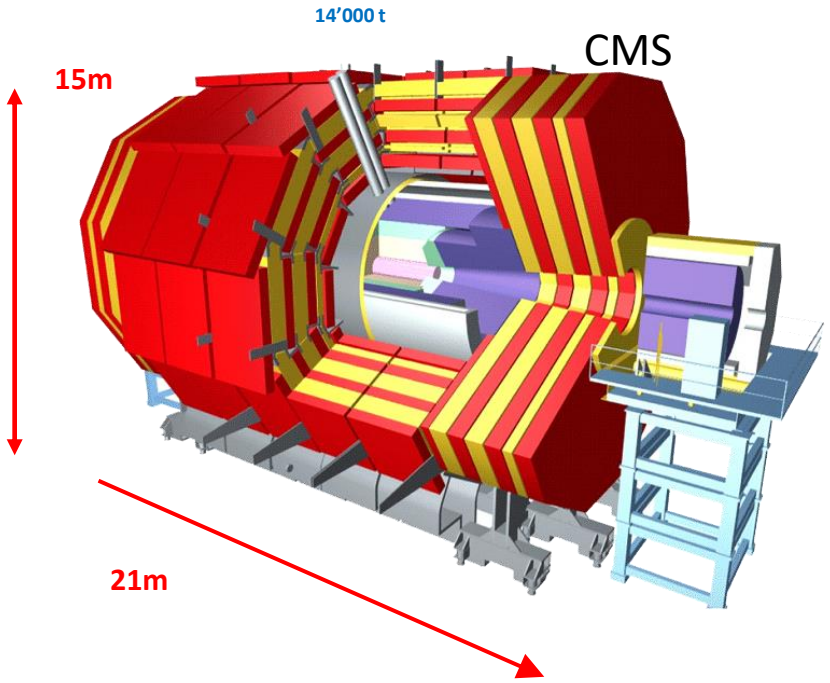
ALICE

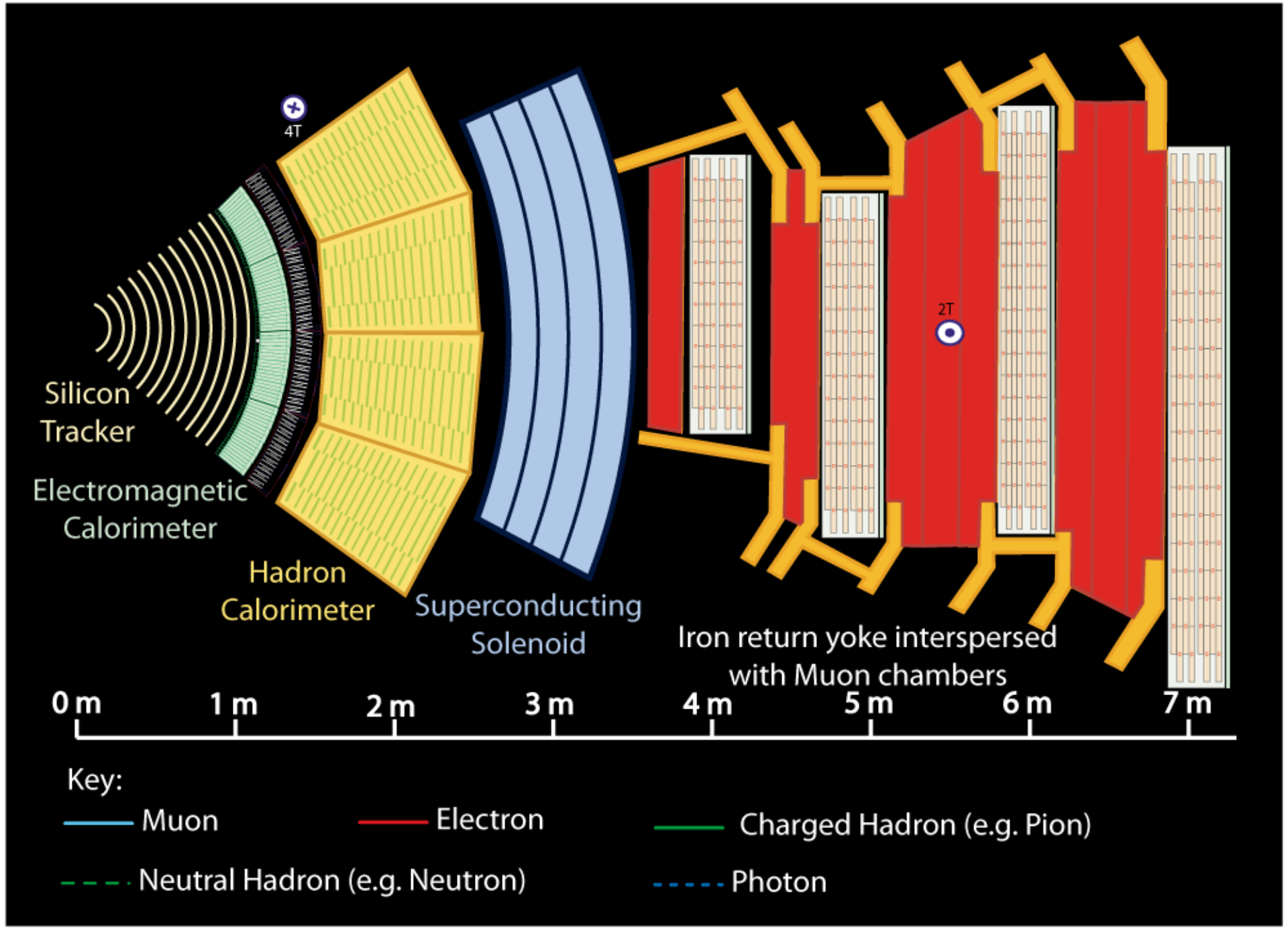


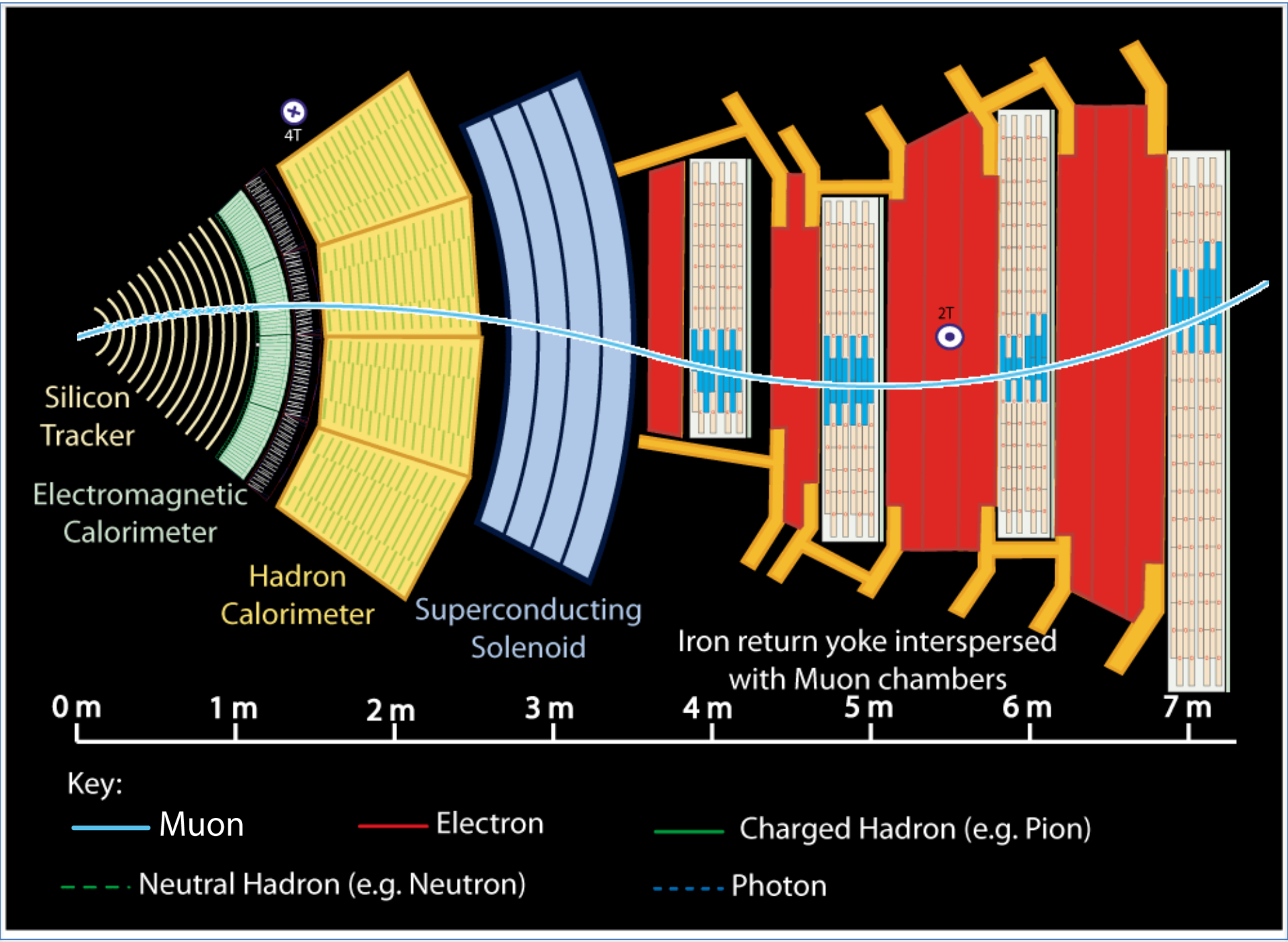
LHCb

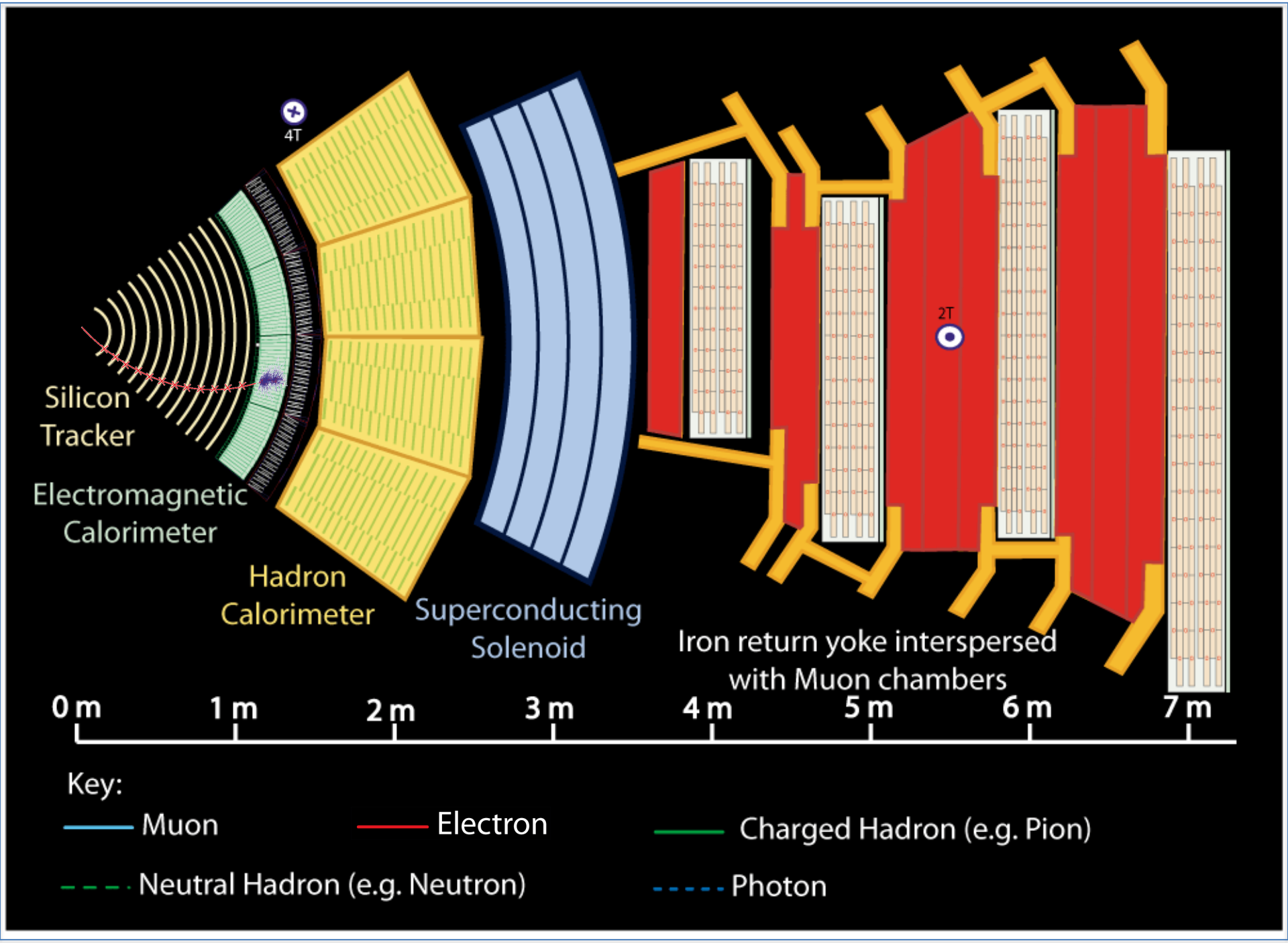


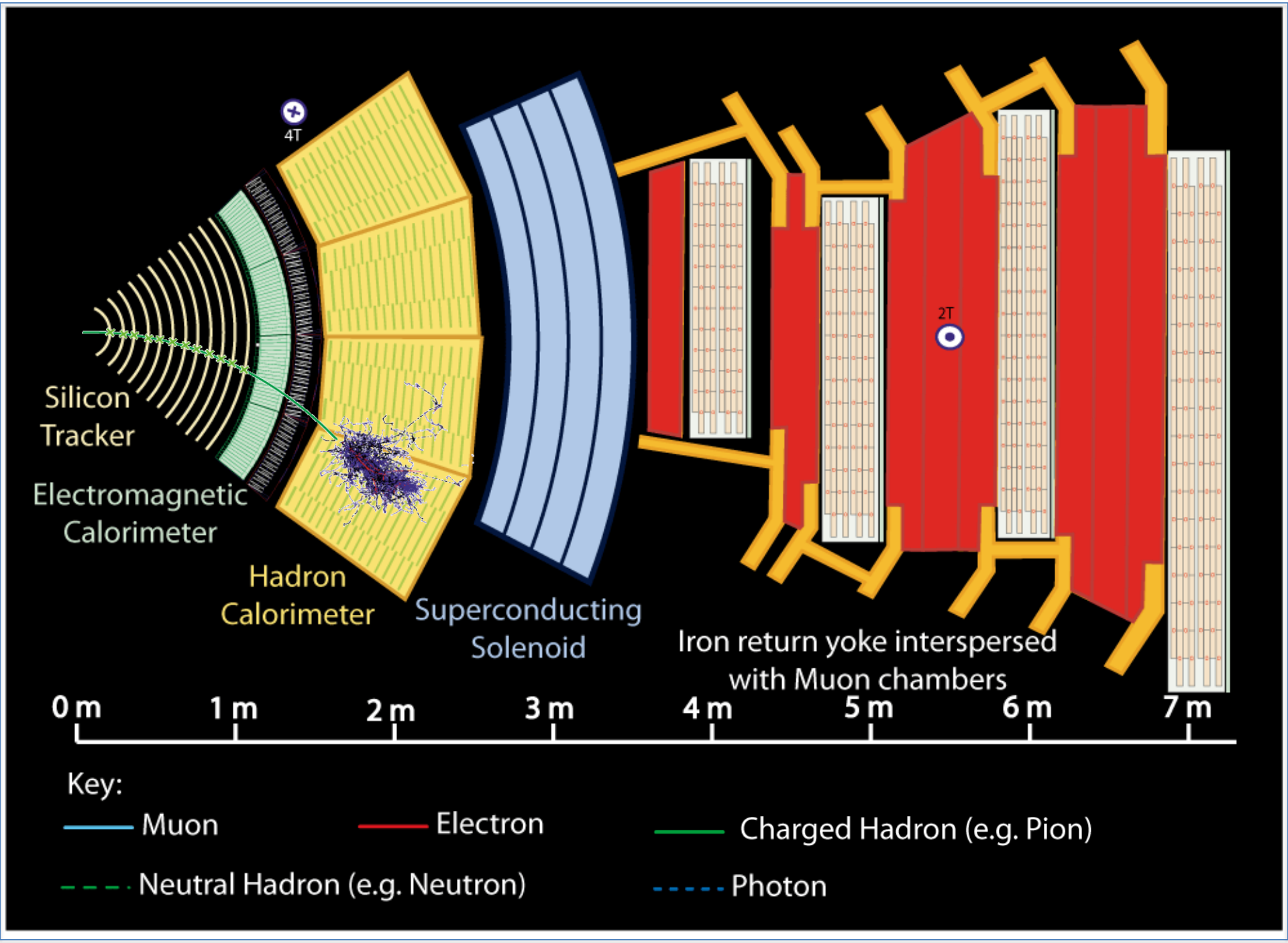
CMS

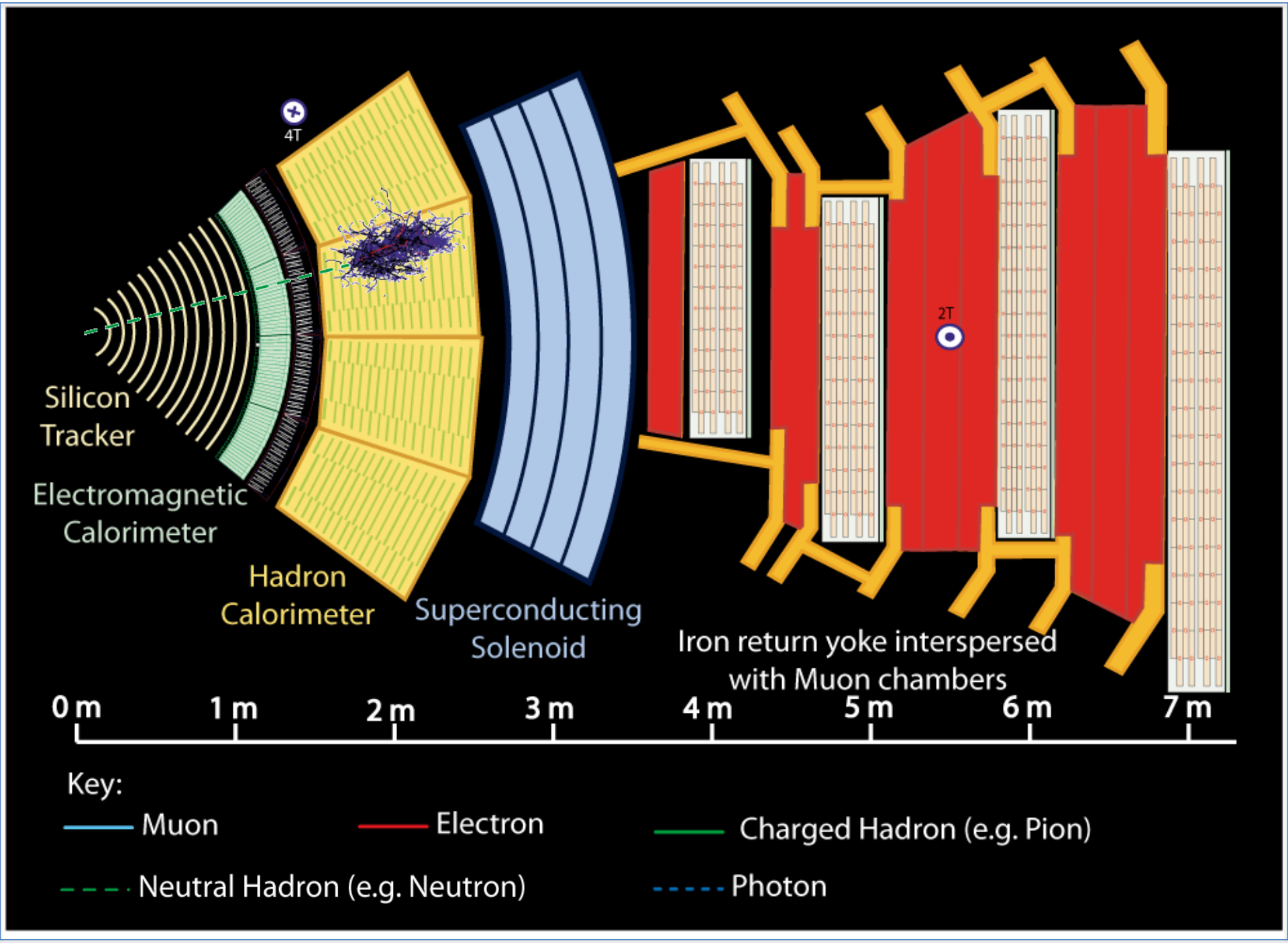


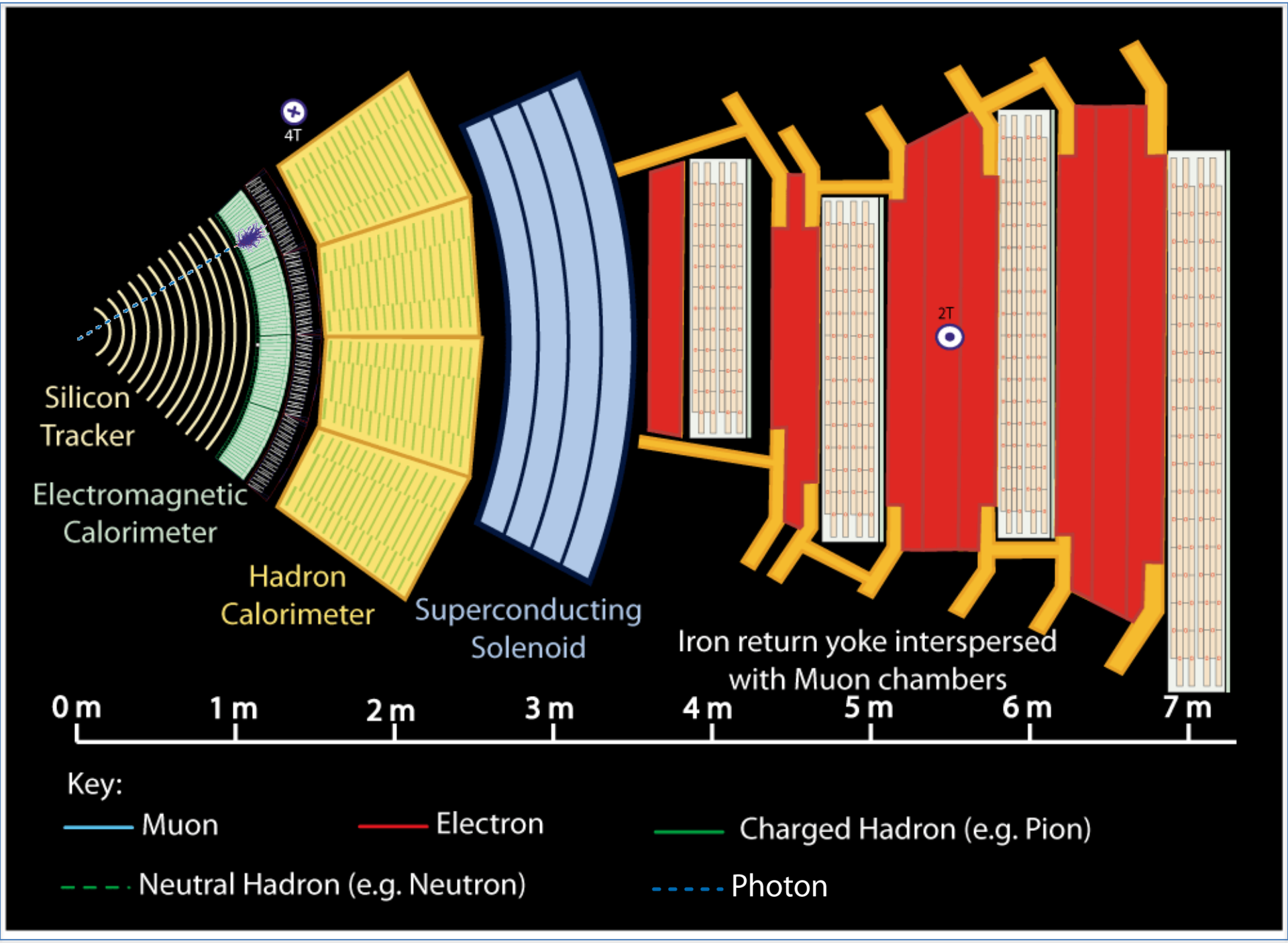

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

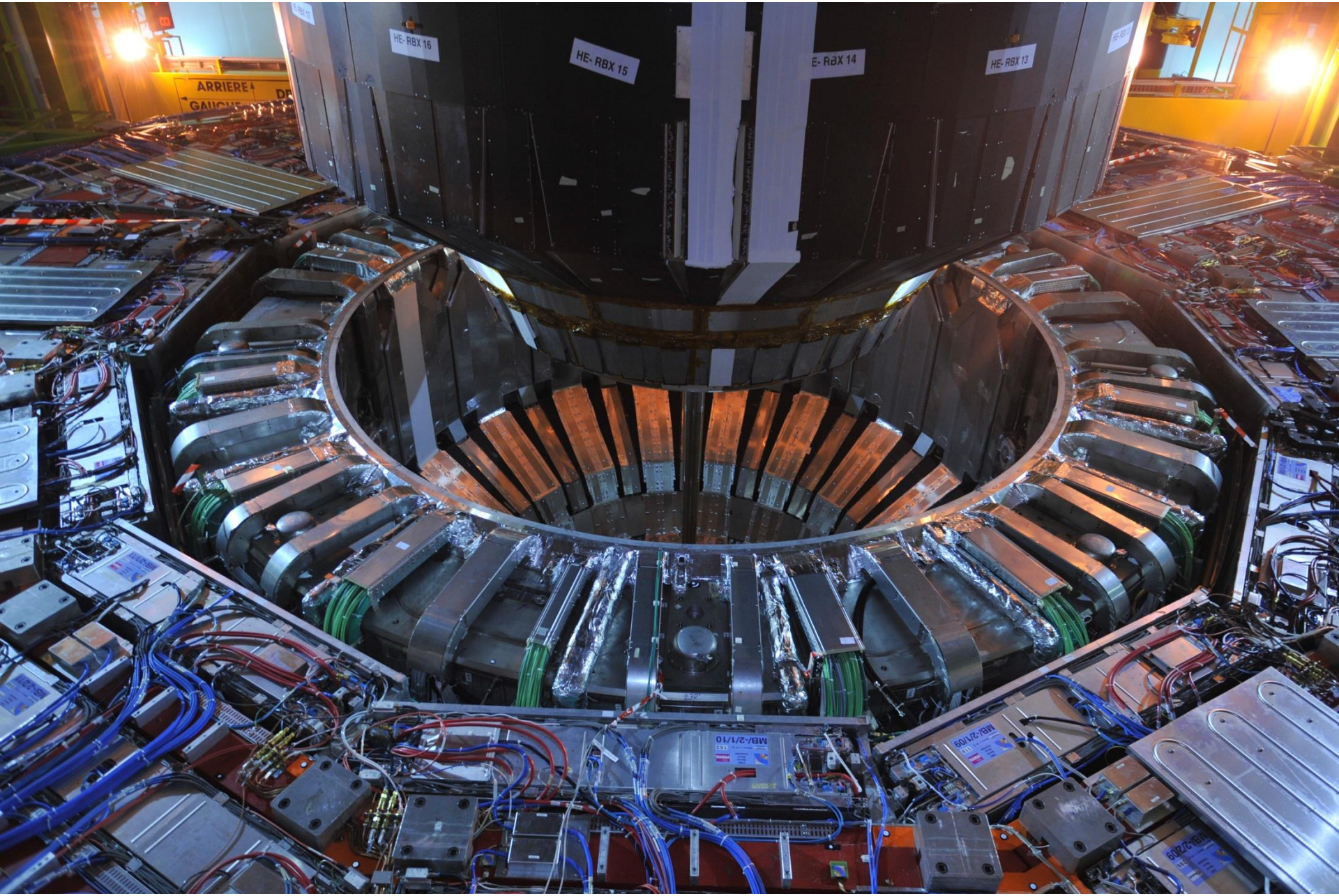

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

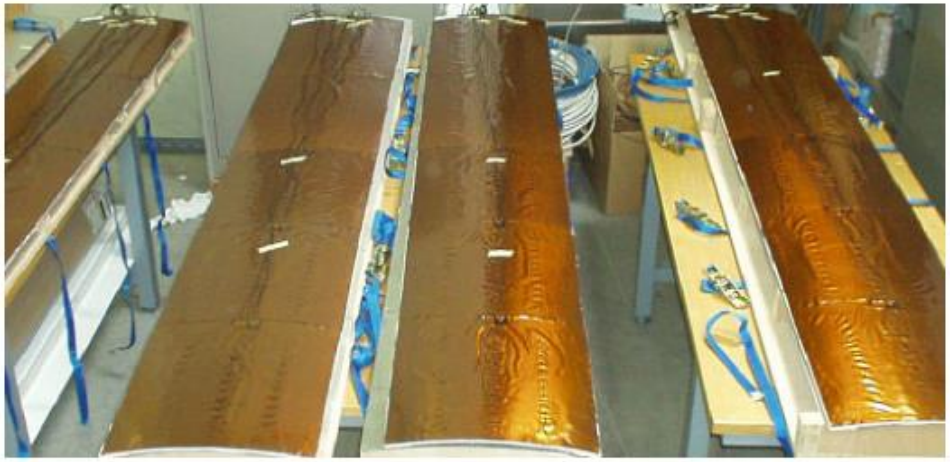

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

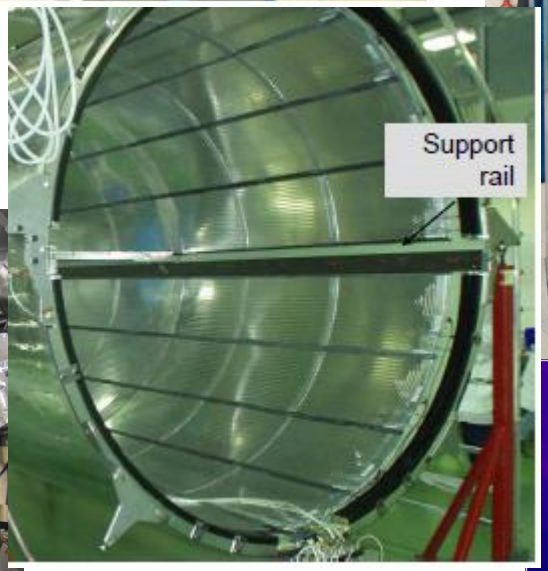

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

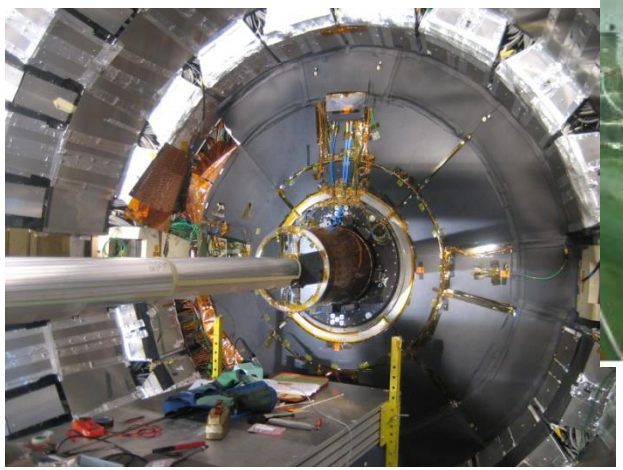

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

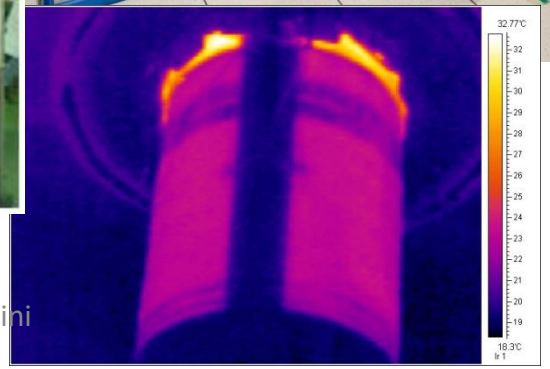

HE- RBX 16
HE- RBX 15
HE- RBX 14
HE- RBX 13
ARRIERE

Συστηματα εξυπνων μονωτων(~ -20 (-40)^{0}C καταγραφεας τροχιων ~ +17(+9)^{0}C ECAL)

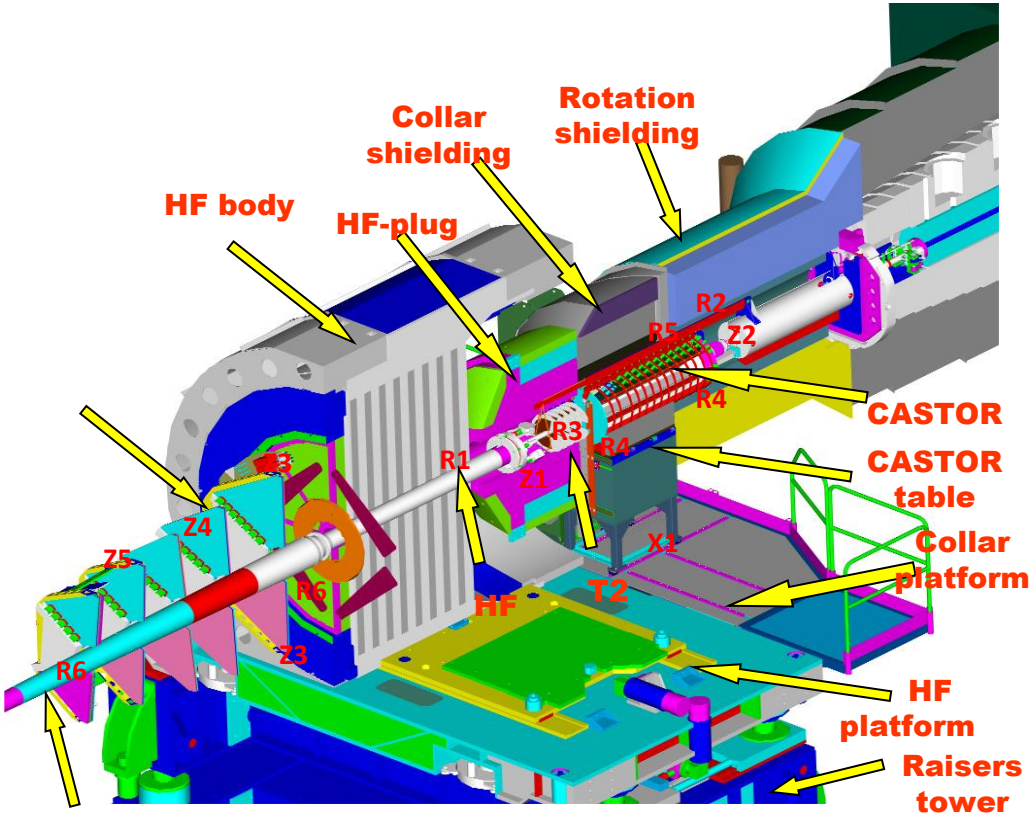






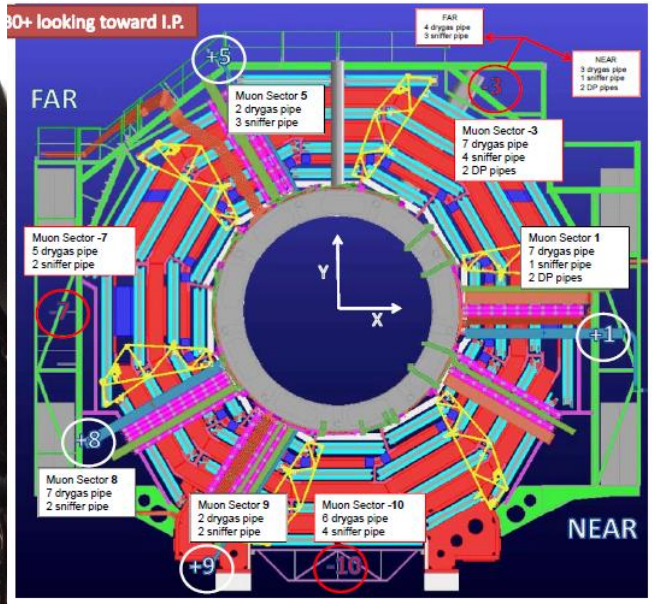


•Συστηματα ελεγχου κινησεων αντικειμενων (υποανιχνευτες και αλλα κοματια “ζυγιζουν” μεχρι 2 τοννους, αισθητηρες)



Τα βαρη των κομματιων μπορουν να αγγιξουν τους 2 τονους και κινουνται με υδραυλικα συστηματα. Ολα αυτα αυτοματα και απο μακρια...μα το προβλημα του να βρεθουν η να αναπτυχθουν αισθητηρες καταλληλοι για τις συνθηκες των πειραματων παραμενει

•Συστηματα ελεγχου περιβαλλοντος (ξερος αερας, αζωτο, θερμοκρασια)



Κοστος ~150,000 ευρω

Προστασια του ανιχνευτη τροχιων απο προβλημα υγρασιας η μαλλον σημειου δροσου

Η πλατφορμα-ecosystem(software+hardware+sensors, κοινοτητα) Arduino https://www.arduino.cc/εχει δωσει τεραστια ωθηση στην μετρηση, σε ολα τα εργαστηρια, τις σχολικες ταξεις και τα ανωτερα ιδρυματα. Το Arduino hardware ειναι προσιτο και απευθυνεται και σε μη ειδικους. Το περιβαλλον προγραμματισμου ειναι πολυ πιο φιλικο και τα εργαλεια που παρεχονται διευκολυνουν την χρηση της πλατφορμας απο ολους. Στην αρχη τα Arduino προγραμματιζονταν σε C++ μεσα στο περιβαλλον Arduino.ide. Αυτο Το περιβαλλον βοηθαει φοβερα στον προγραμματισμο αλλα το προβλημα της γλωσσας παραμενει διοτι η C++ δουλευεται απο ειδικους μονον. Οταν ομως τα Arduino αρχισαν να χρησιμοποιουνται στα σχολεια υπηρξαν διαφορες προσπαθειες για απλες γλωσσες προγραμματισμου (graphical –Scratch και Blockly). Οι γλωσσες αυτες δεν μπορουν να χρησιμοποιηθουν σε ολο το φασμα δυνατοτητων του Arduino και απαιτουν πολλες φορες ειδικες εκδοσεις του hardware. Επισης, η εκμαθηση τους για μαθητες που προσβλεπουν σε Τεχνικο/επιστημονικο μελλον προσφερουν μια κακη προπαιδευση.

Στο μεταξυ, η γλωσσα προγραμματισμου Python (https://micropython.org/) **εκανε την επανασταση (μια ακομα!) στον χωρο προγραμματισμου, με διαρκως περισσοτερους οπαδους-εφαρμογες. Στην Python (διερμηνευμενη γλωσσα) ο κυκλος προγραμματισμου ειναι πολυ πιο συντομος και αμεσος.**

Η Python μεταφυτευτηκε στο Arduino hardware σαν μPython και ειναι μια γλωσσα που παρεχει την δυνατοτητα πληρους αξιοποιησης της πλατφορμας και παρα πολλα ηλεκτρονικα κομματια παρεχουν το software ετοιμο. Επιπλεον ειναι η ιδανικη γλωσσα να επιδειξουμε σε μαθητες που εχουν τεχνολογικες/επιστημονικες ανησυχιες.

Το Arduino και το προγραμμα LED blink 

Η πραγματικη επανασταση του συστηματος Arduino ειναι ολα τα επιπεδα αφαιρεσης αναμεσα στην δουλεια της CPU και στα ηλεκτρονικα κομματια , (I/O pins).

Where before you would need to manipulate the registers with cryptic instructions such as

```
DDRD = 0b11111111; // declaring port D as output, because we know the LED is connected to port D, pin 0
PORTD |= 0b10000000; // pin 0 of port D set HIGH, all other pins left unchanged, this is an OR operation
_delay_ms(1000); // delay of one second
PORTD &= 0b01111111; // pin 0 of port D set LOW, all others left unchanged, this is an AND operation
```

now you can achieve the same result with more meaningful and understandable instructions:

```
pinMode(LED_BUILTIN, OUTPUT); // we do not need to remember which port and pin the LED is connected to
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level), no need to manipulate bits
delay(1000);                                  // wait for a second
digitalWrite(LED_BUILTIN, LOW);   // turn the LED off (LOW is the voltage level), again no bit manipulation needed
```

Αναλυση προγραμματος

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}


void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000
}
```

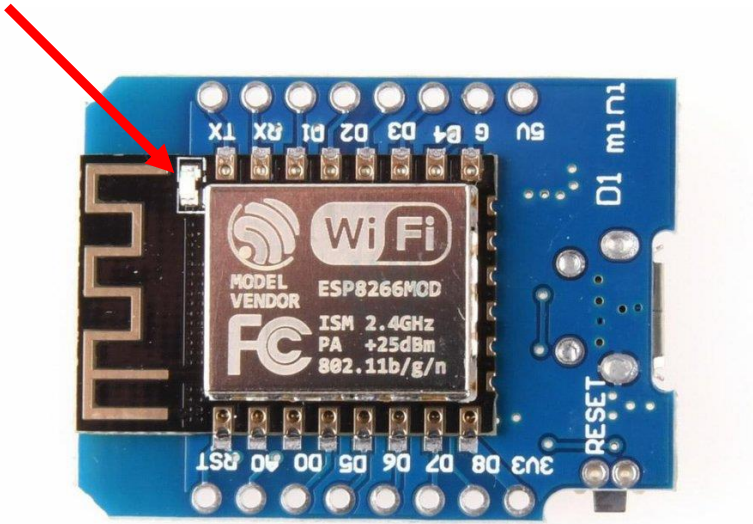
The setup() function is executed only once, at the beginning of the program. In this case, it handles the mechanisms needed to make the pin the on-board LED is connected to an output (so we can turn the LED on and off)

The loop() function, instead, is supposed to keep running "forever". In this case, it makes the LED pin take a HIGH value (+5 V) so the LED turns on, waits for 1 second, makes the LED pin take a LOW (0 V) value so the LED turns off, then waits 1 more second before ending and restarting again, and again, and again...

Το ιδιο "blinky" παραδειγμα σε MicroPython

- We can now try the LED blink example in MicroPython: in Thonny, create a new file (click on the blank page icon) and type in the following, being very careful to respect the indentation (use the Tab key "-->"):

```
import time
from machine import Pin
ledpin = machine.Pin(2, Pin.OUT)
while(True):
    ledpin.on()
    time.sleep(0.5)
    ledpin.off()
    time.sleep(0.5)
```



Τι συμβαινει στο"blinky" προγραμμα;

```
import time
from machine import Pin
ledpin = machine.Pin(2, Pin.OUT)
while(True):
    ledpin.on()
    time.sleep(0.5)
    ledpin.off()
    time.sleep(0.5)
```

These two lines are similar to #include <library.h> in C/C++

here we define the LED pin

now we loop forever

turn the pin to 1 (LED off)

wait half a second

turn the pin to 0 (LED on)

wait half a second

Οι αισθητηρες που θα χρησιμοποιησουμε μετρανε ο,τι χρειαζομαστε και με τους ανιχνευτες

- **Θερμοκρασια**
- **Αποσταση**
- **Υγρασια**
- **ροη**
- **πιεση**
- **Και ο,τι αλλο….**

Η διαφορα ειναι οτι εσεις μπορειτε να χρησιμοποιητε ψηφιακους αισθητηρες... εμεις ΟΧΙ.
Παρακατω βλεπετε αισθητηρες που χρησιμοποιουνται καθημερινα απο ολους μας....

Ο αισθητηρας BMP180 (θερμοκρασια, πιεση)

3V3
G
D1
D2

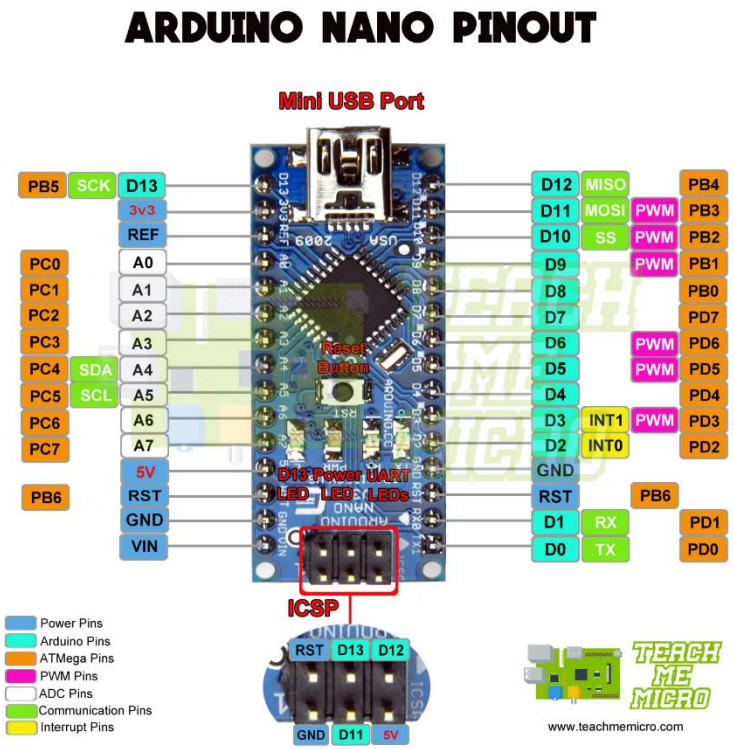
BMP180 (Bosch):

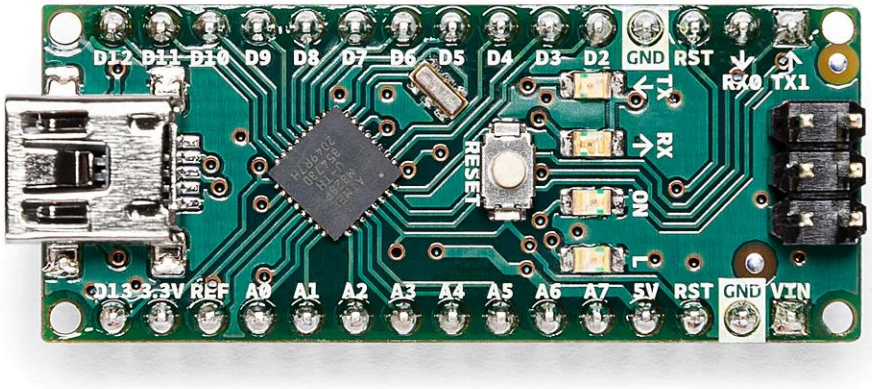
- VIN: power voltage, 3.3 V - pin "3V3" on CPU board;
- GND: power return, GND - pin "G" on CPU board;
- SCL: input I2C clock line (ρυθμος επικοινωνιας!)- pin "D1" on CPU board, GPIO 5;
- SDA: input/output I2C data line - pin "D2" on CPU board, GPIO 4;
- measures Temperature between 0 and 65 C, pressures between 300 and 1100 hPa (corresponding to +9000 and -500 m relative to sea level)

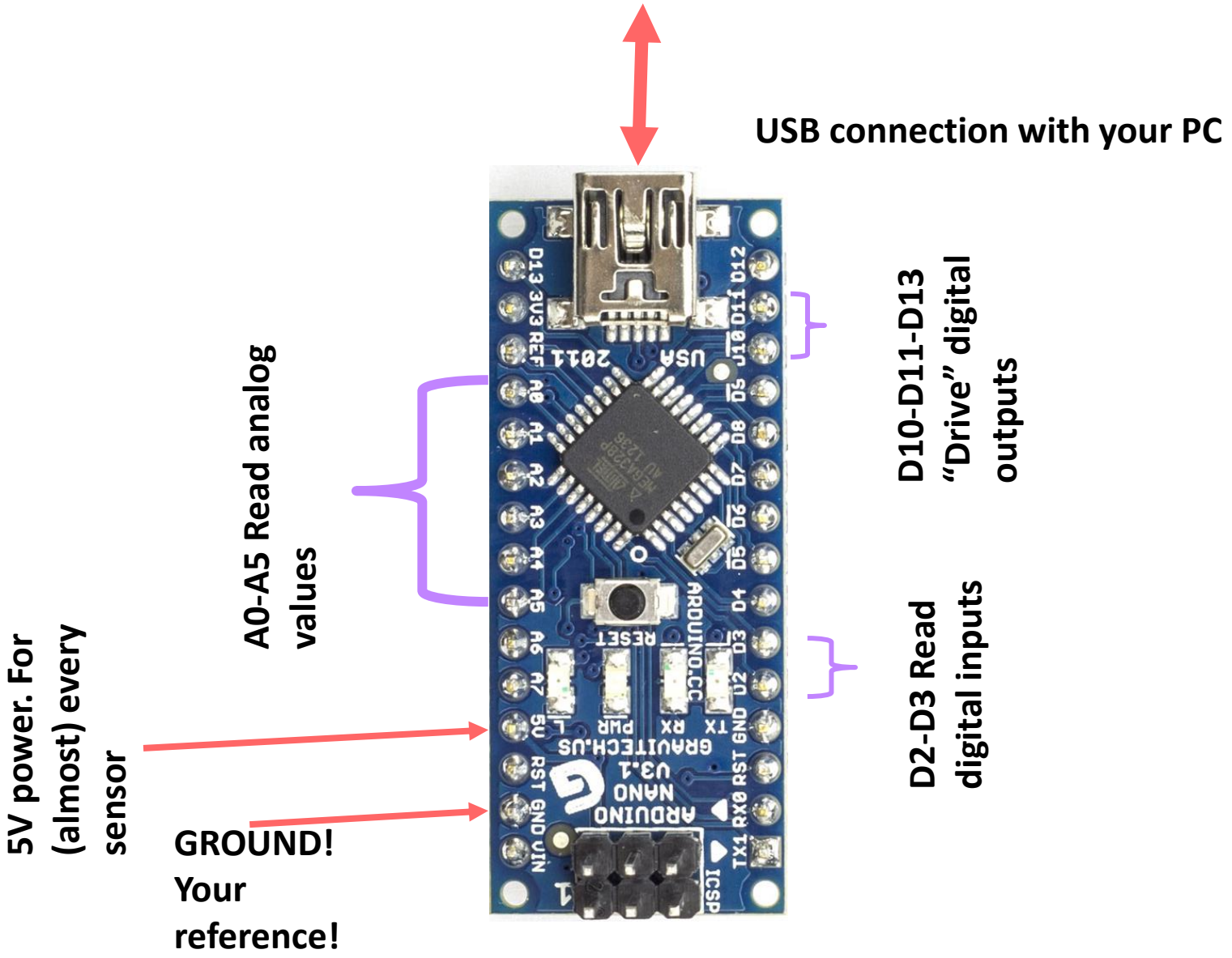
Arduino βιβλιοθηκες για αισθητηρες! Πολλες!

Another great strength of the Arduino "system" is the availability of "libraries" to handle many, many types of hardware. From the Arduino IDE, under "Tools", select "Library Manager" and enter, for example, BMP180. Scroll down until you find "BMP180MI" and click on "Install". After a short while, you will have this library available to call in your programs. Most libraries also come with example programs you can open, read, modify and play with. The BMP180 is not an exception. Similarly, you can search for AHT20 and install a library for the AHT20 (and AHT21) sensor, and an example program, or search for LM75A to get a library for the LM75A sensor, with examples.

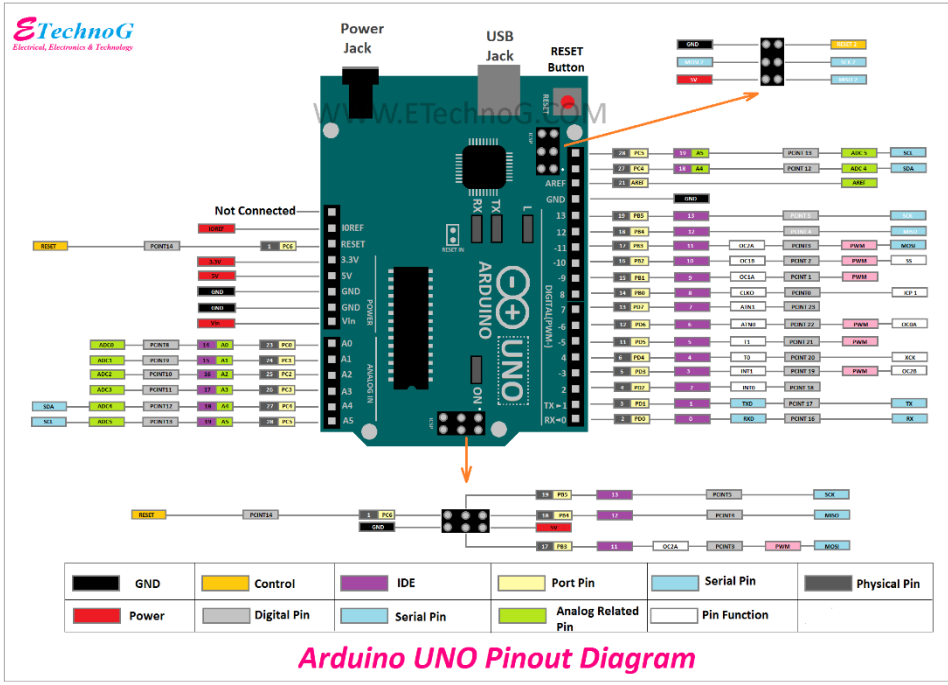
Arduino Nano pinout: which pins can do what



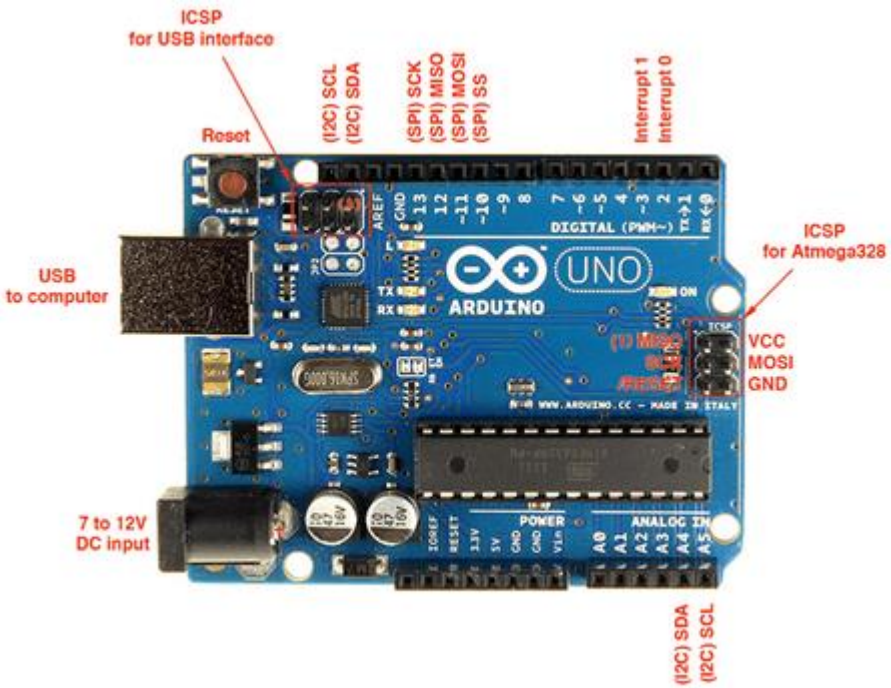



USB connection with your PC
A0-A5 Read analog values
D10-D11-D13 "Drive" digital outputs
D2-D3 Read digital inputs
5V power. For (almost) every sensor
GROUND! Your reference!

Arduino UNO pinout



Arduino UNO Pinout Diagram

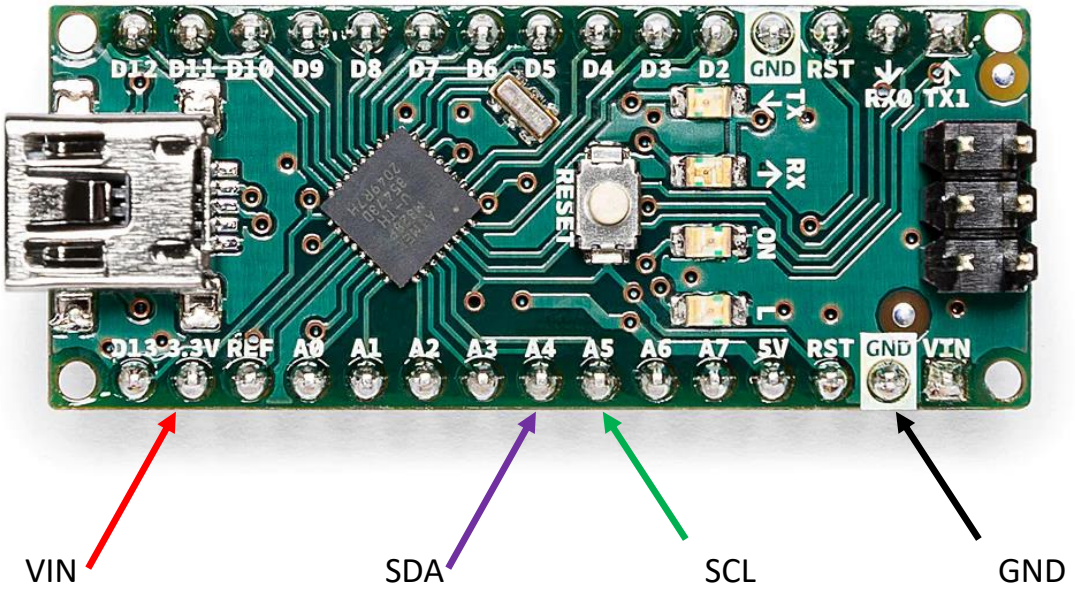


How do we connect our sensors? BMP180

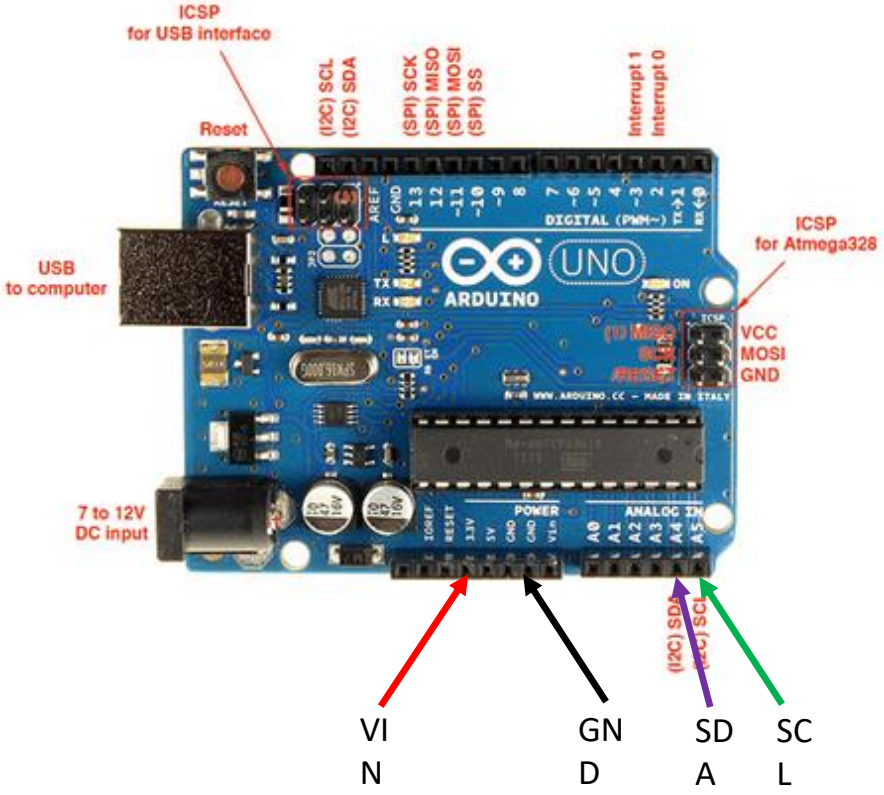
The BMP180 sensor board has four pins. We can connect them to our Arduino Nano as follows:

- VIN is the power supply for the sensor, which MUST be 3.3V and NOT 5V (otherwise you will destroy it). So, we need to connect it to the "3v3" pin of the Nano.
- GND is the power return, and it goes to one "GND" pin of the Nano (there are two available).
- SCL is the Serial CLock signal needed to read the sensor, and it needs to be connected to the "SCL" pin of the Nano, which is also called and labeled "A5".
- SDA is the Serial DAta line used to read the sensor, and it needs to be connected to the SDA line of the Nano, which is also called and labeled "A4".

Connections, in detail



Arduino UNO version of the connections:



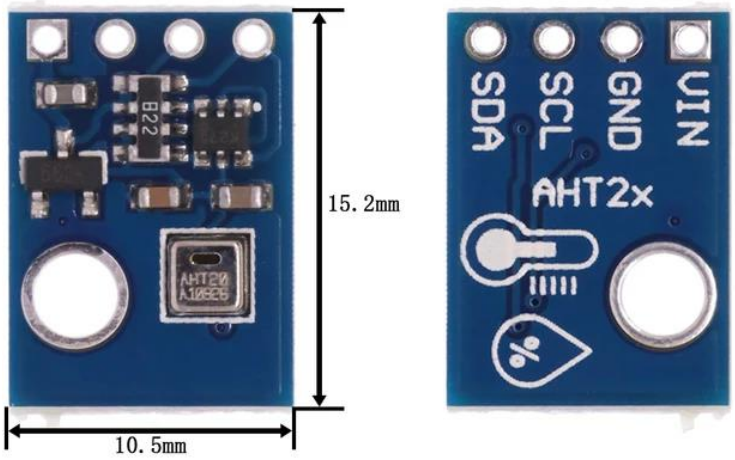
You can totally ignore the explanations for the pins you are not interested in, or check them out for the future...

Note that the pins are located in different positions.

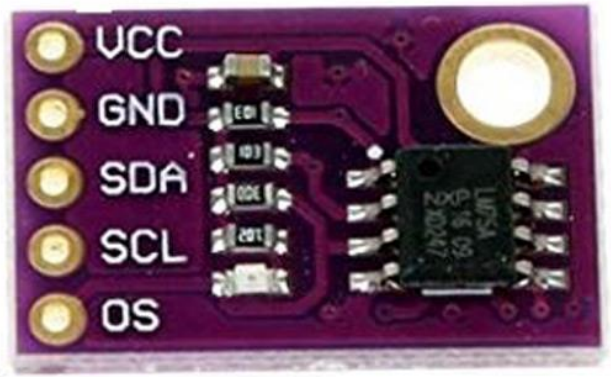
Connecting the AHT21 sensor board (αισθητηρας υγρασιας+θερμοκρασιας)

The AHT21 board also has four pins, in fact they are the same as for the BMP180 board but are not placed in the same positions, rather the other way around, so please be careful. You can disconnect the Arduino UNO from USB, disconnect the BMP180 board one wire at a time and reconnect each wire to the AHT21 immediately.

Write down which color wires you are using for which signal, and try to keep the association between signal and wire color always the same, and you will prevent trouble.



Connecting the LM75 board



The LM75 board seems to have one more connection, the one labeled "OS", but we do not really need it. It is an output that the LM75 makes active whenever the temperature exceeds a set limit (so it can be used as a thermostat).

In fact, ignoring "OS", the connections are similar to those for the AHT21, except for the exchange of SCL with SDA (this is an annoying fact, but there is no "standard" way of connecting these boards, as every producer feels free to locate the pins wherever it is more convenient. For him, not for us).