LECTURE 1

1.1 What is IoT (Internet of Things)

This course is going to explain what IoT represents, what it consists of, also name some of the most important hardware which make it happen and try to give IoT a proper understanding of its meaning and importance.

The Internet of Things has become a very widely spread concept in the last few years. The reason for this is mainly **the need to computerize and control most of the surrounding objects and have access to data in real time**. Think about parking sensors, about phones which can check the weather and so on.

The Internet of Things represents the whole way from **collecting data**, **processing it, taking an action corresponding** to the signification of this data to **storing everything in the cloud**. All this is made possible by the internet.

An example are sensors, they collect data and send it to a processing device, which will perform the convenient actions. Then, the data will be stored locally and, by using the internet, it is subsequently sent out to the cloud. The problem here is that the data stored in the cloud is sometimes not useful. There is not enough local processing happening before data is saved in the cloud.

The ideal scenario towards which the Internet of Things is headed, would be to have a computer store data locally, process it, check for abnormalities or search for relevant segments and upload only this information to the cloud. This concept is called either Edge, according to Intel, or fog computing as stated by Cisco, implying what happens before the cloud.

The IoT is involved in medicine, agriculture or transportation due to sensors and cloud storage. *(HomeWork :Give2 exampels of IoT in each sector)*

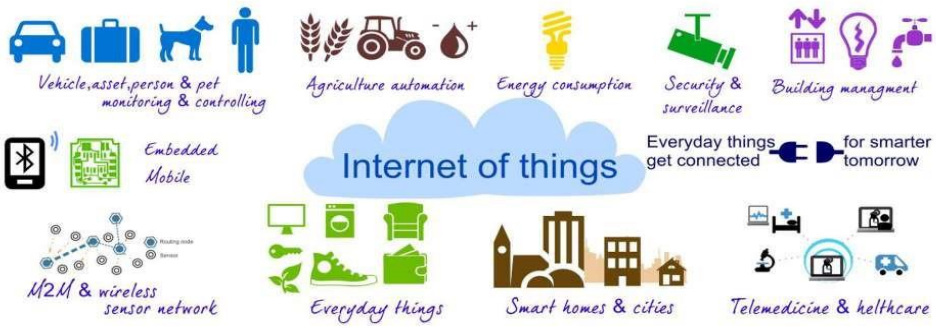

Figure 1 Applications of IoT in life Sectors

1.2 IoT Stack (IoT components)

A basic Internet of Things system consists of the following components:

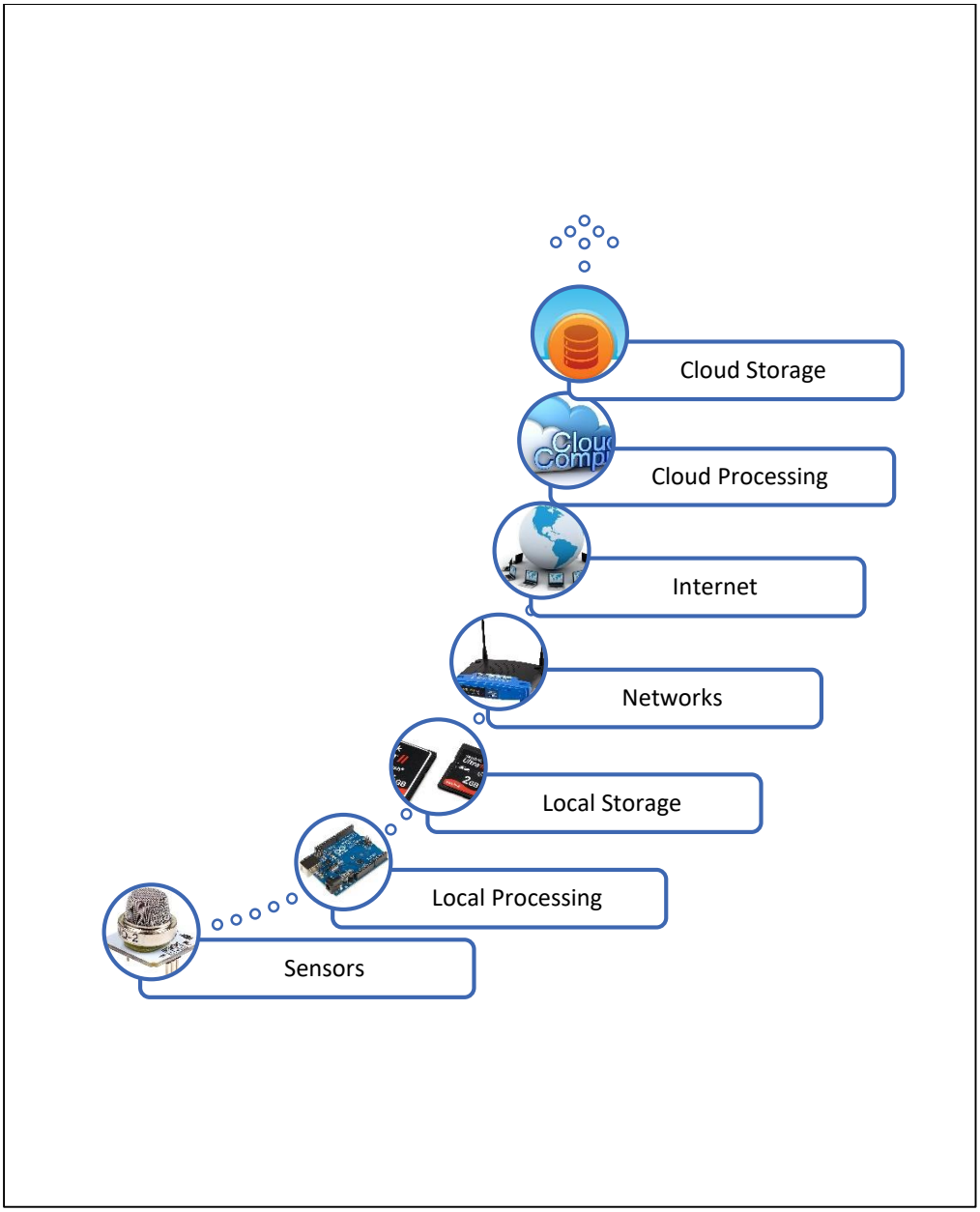

Figure 2 IoT Stack

LECTURE 2

1.2.1 Computers

Conversely, since it is a computer, the **Raspberry Pi** runs an Operating System. That means that you can run multiple programs on it and you can run applications that use Internet services. However, this also implies that the application you run on the Raspberry Pi is not real time, thus you cannot estimate when a certain sequence will get executed.

Raspberry Pi came second with the qualities of being cheap and useful. It is actually a small computer, it runs Linux as operating system and has a full network system, solving thus the processing problem of the Arduino.

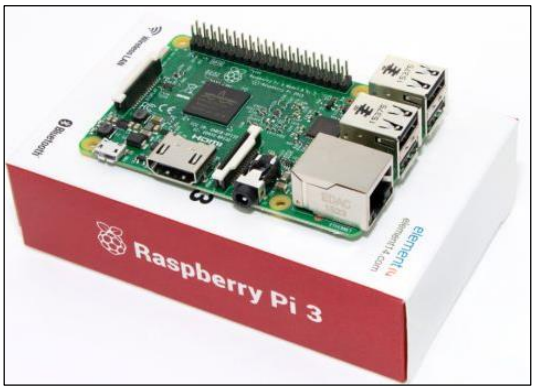

Figure 3 Raspberry PI3

1.2.2 Development devices

The appropriate hardware can be found by checking many aspects. There are many boards, but they are good for different kinds of projects.

- **For sensor handling**

  Arduino, ChipKIT and LounchPad are the best options.

- **For processing**:

  the **STM32** with 128KB of RAM, **Particle** with an ARM chip and wifi, **Espruino** which is actually a JavaScript machine and so, you can write JavaScript code to run on it.

- **For processing and network**, there are the **Raspberry Pi**; **Intel Galileo** which handles hardware pretty good, has 256MB of RAM and a Qark processor; The Intel Edison which comes as an improvement to the Galileo, can be used with wifi and bluetooth and has a 4GB flash memory; **Beaglebone Black** also brings flash memory; **UDOO Neo** is a combination of a Raspberry Pi and an Arduino, **Parallella** which is very different from the others, being a prototyping board, plus, it has a chip which processes 16 or 64 programs at a time

1.3 Software

Prototyping is a necessary process while building a professional product. The software used for it should be fast to write, easy to deploy when finished and it's there to make a proof of concept. However, it is not supposed to be user level grade.

- **Prototyping** has the only property of making a statement, whereas professional programming can't have any faults, can't break and has to meet the client's needs. The softwares built for this job are Eclipse, VIM, MBED- online platform where you can write the code and download the binary file for the board- ,Intel XDK which uses JavaScript as language for developing and also offers HTML as an alternative.
- The field of **data acquisition and analysis** brings:**Xively**, which collects data and displays it, offers libraries for you to integrate into your project, send the latter to the cloud. Here, you can add graphs and monitor everything.**Microsft Azure**, through which you push data to the cloud with failure detection as target. With this software, there's no need for machine learning algorithms when you can use Microsoft's technology and knowledge to build prediction models.
- **Solution builders** such as Wyliodrin STUDIO allow prototyping and build solution projects to be sent to the clients.