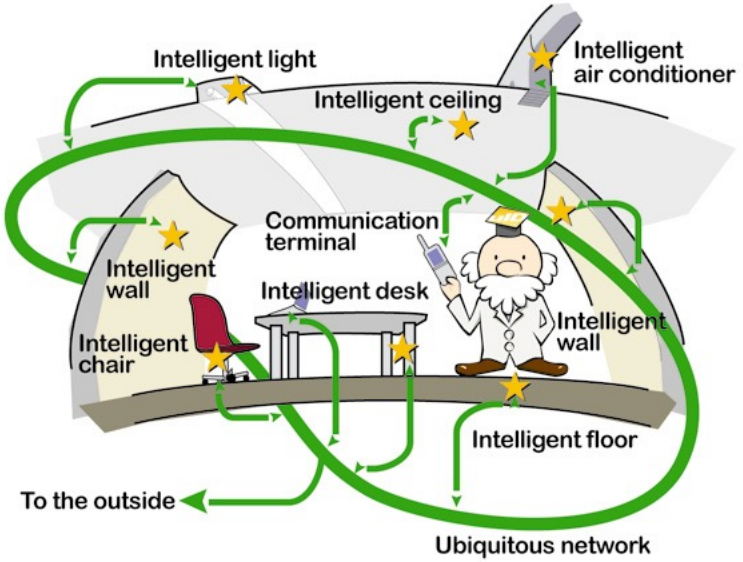
Ubiquitous Computing and Embedded systems

Fernando Guirado
Office 3.17
EPS building
fernando.guirado@udl.cat



Let’s start

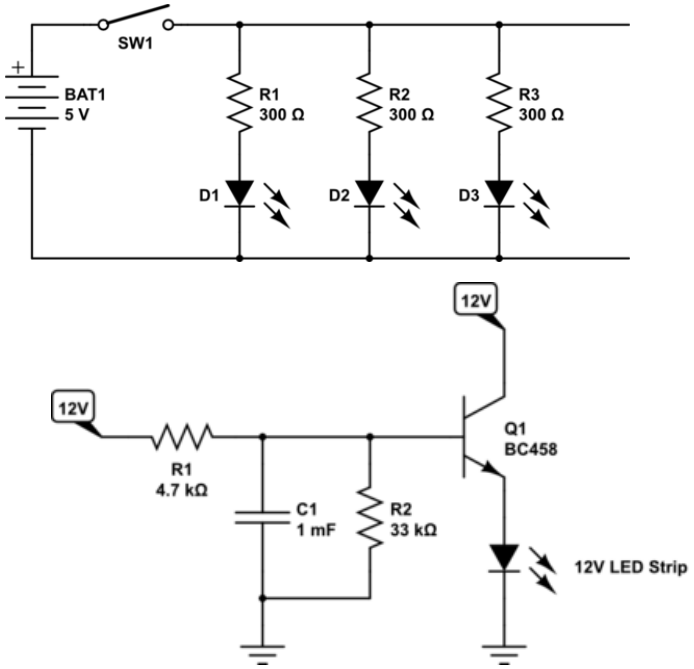
Some questions

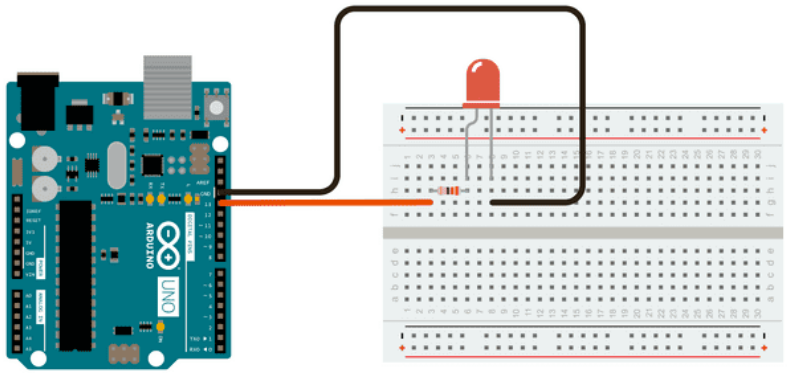
1. C/C++ language programming?
2. Microcontrollers? Arduino, RPi, PIC, STM, ESP, ….

Let’s start

Some questions

1. C/C++ language programming?
2. Microcontrollers? Arduino, RPi, PIC, STM, ESP, ….
3. Electronic skills? Resistors, capacitors, voltage, current, …

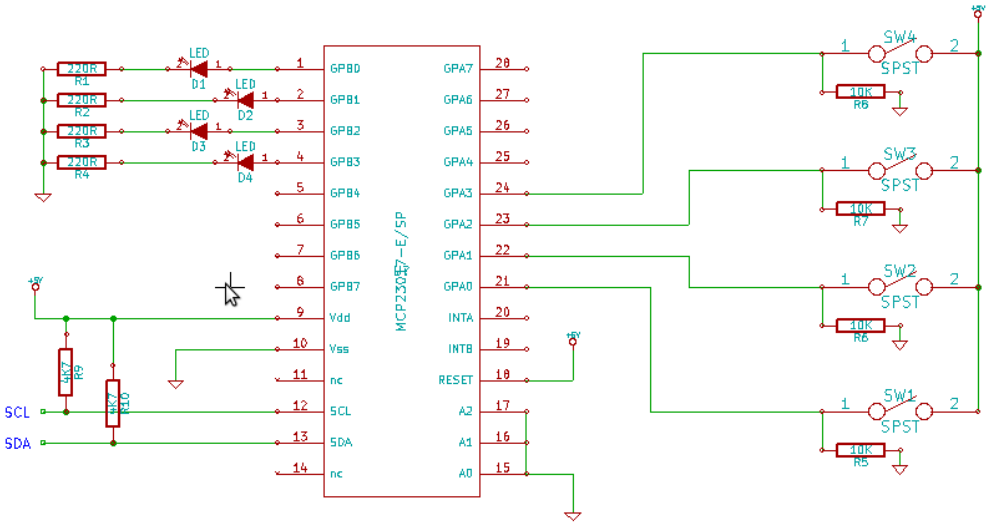


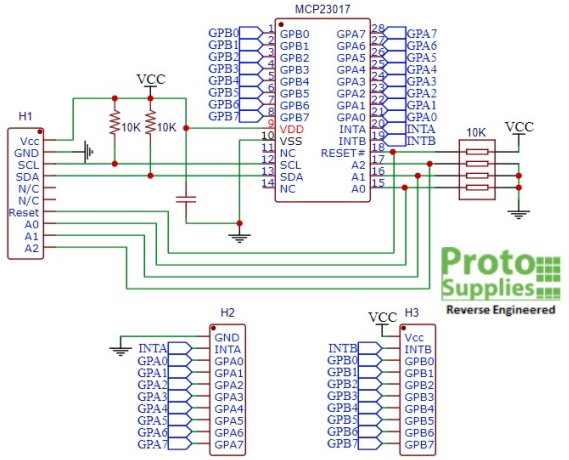


Let’s start

Some questions

1. C/C++ language programming?
2. Microcontrollers? Arduino, RPi, PIC, STM, ESP, ….
3. Electronic skills? Resistors, capacitors, voltage, current, …

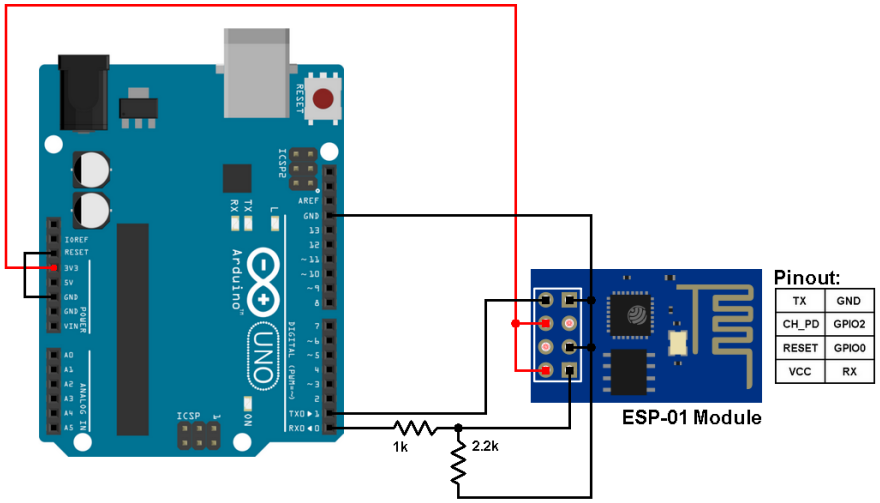


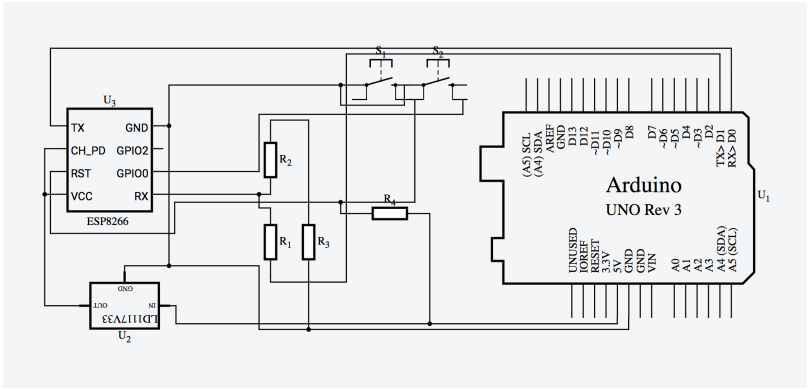


Let's start

Some questions

1. C/C++ language programming?
2. Microcontrollers? Arduino, RPi, PIC, STM, ESP, ….
3. Electronic skills? Resistors, capacitors, voltage, current, …



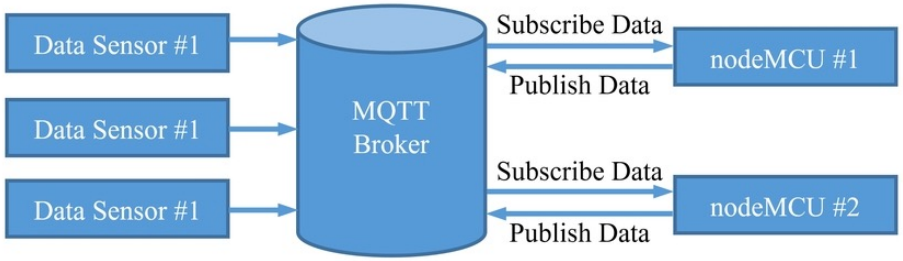


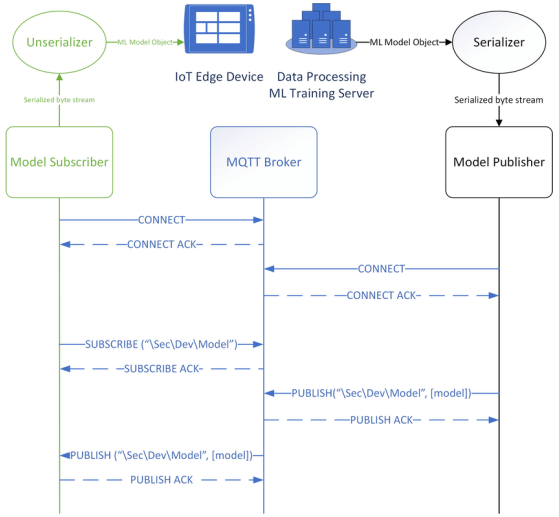
Let's start

Some questions

1. C/C++ language programming?
2. Microcontrollers? Arduino, RPi, PIC, STM, ESP, ….
3. Electronic skills? Resistors, capacitors, voltage, current, …
4. Network management and programming?

   IP address, STA vs AP, protocols, …





Let's start

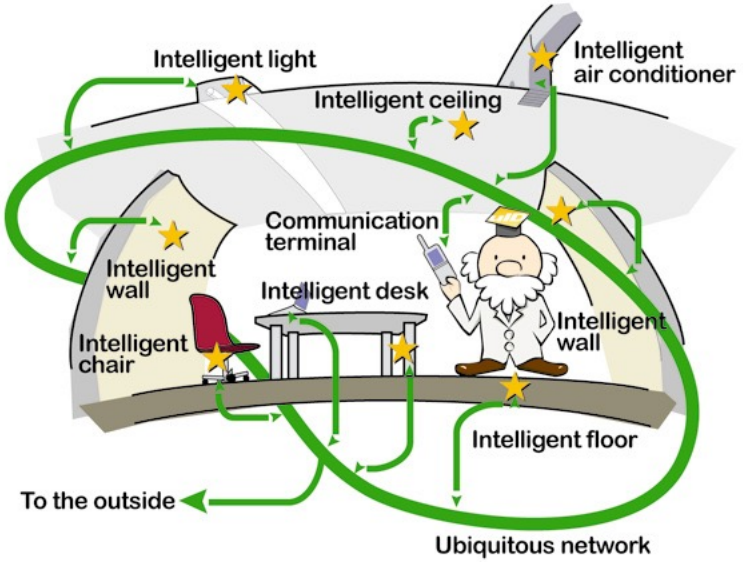
Some questions

1. C/C++ language programming?
2. Microcontrollers? Arduino, RPi, PIC, STM, ESP, ….
3. Electronic skills? Resistors, capacitors, voltage, current, …
4. Network management and programming?

   IP address, STA vs AP, protocols, …
5. RTOS?
6. IoT?
7. Cloud/Fog/Edge Computing?

Ubiquitous Computing and Embedded systems

Fernando Guirado
Office 3.17
EPS building
f.guirado@diei.udl.cat



Competences

- Master Information and Communication Technologies
- Master a foreign language
- Capacity to design and develop computer systems, applications and services to built-in and ubiquitous systems
- Capacity to draft, design and implement projects and/or give novel solutions, using engineering-related tools

Course structure

- Little theory presentation on the main topics
- Short exercises to get practice
- A development project based on SCRUM

Contents

Embedded Systems & RTOS

Internet of Things

Edge Computing

Contents

Embedded Systems & RTOS

Internet of Things

Ubiquitous Computing

Contents

Embedded Systems

1. Introduction
2. Embedded Systems Characteristics
3. Functional and timing requirements
4. Real Time: Modelling, Scheduling, Design

Real Time Systems

- Real Time Systems are designed taking into account both; hardware and software.
- It processes any external/internal stimuli under highly tight timing constraints.
- RTOS allows the hardware access while meeting the timing constraints.

Contents

Embedded Systems & RTOS

Internet of Things

Ubiquitous Computing

Contents

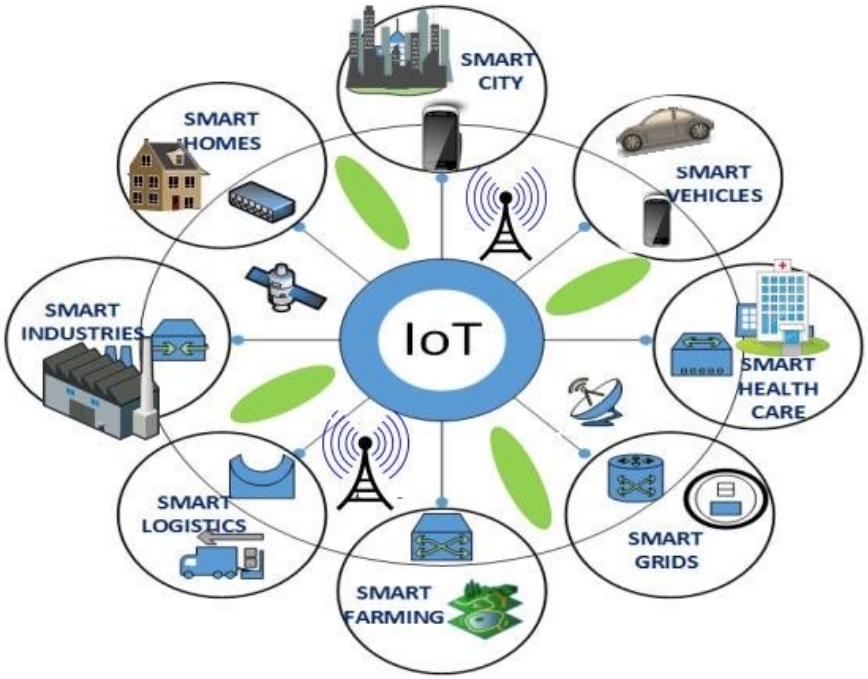
Internet of Things

The **Internet of Things** (**IoT**) is the network of physical objects (things) - devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity - that enables these objects to collect and exchange data over the Internet.

IoT

Big Challenges

- Data Security. Some IoT devices collect highly sensitive information.
- Data Privacy. Much of the information collected and processed by IoT devices may be protected under various data privacy laws. ...
- Data Volume. ...
- Data Complexity.



Contents

Internet of Things

1. Introduction
2. Components of IoT and technological restrictions
3. Connectivity and communication protocols

Contents

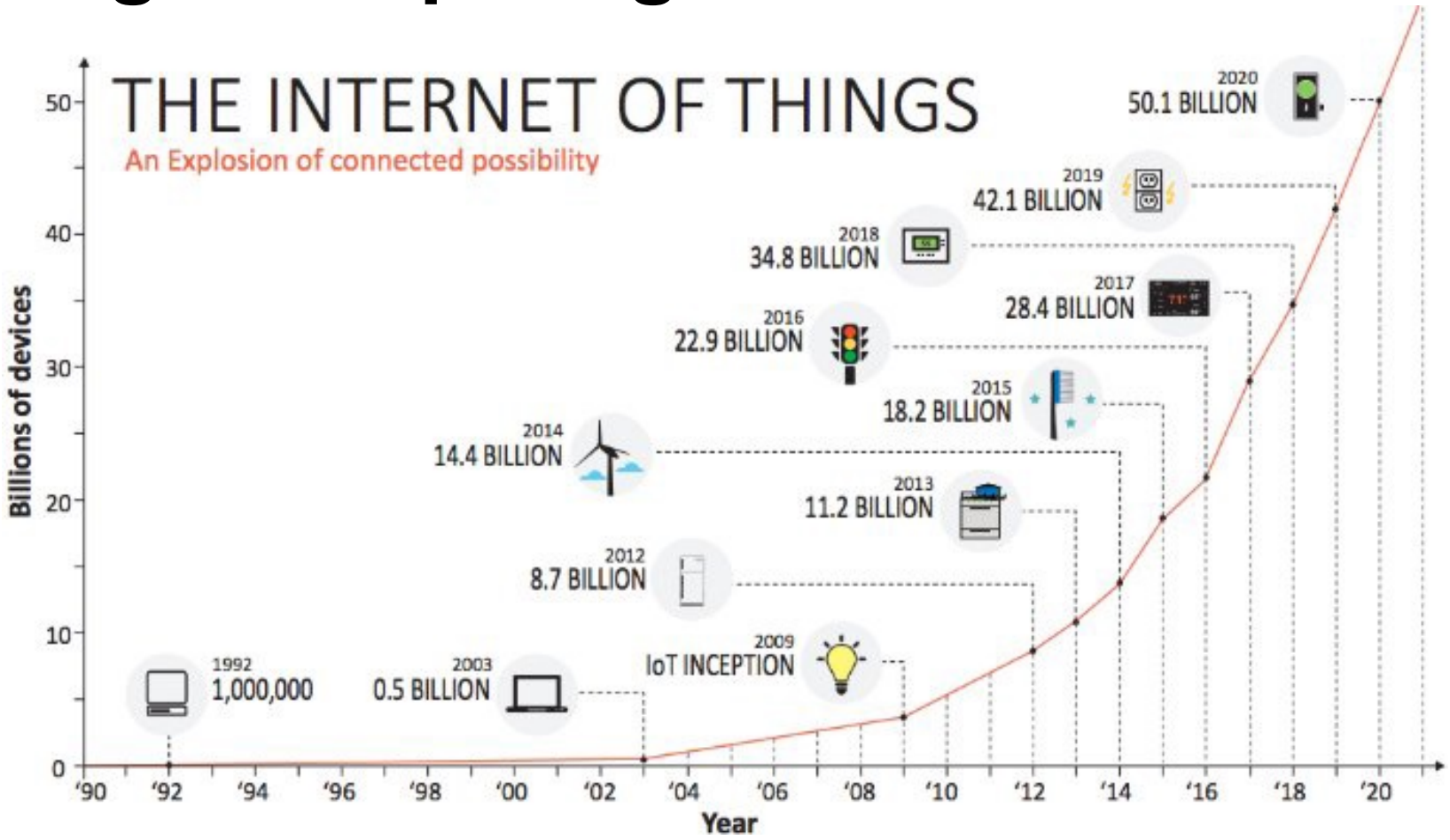
Embedded Systems

Internet of Things

Edge Computing

Contents

Edge Computing



Contents

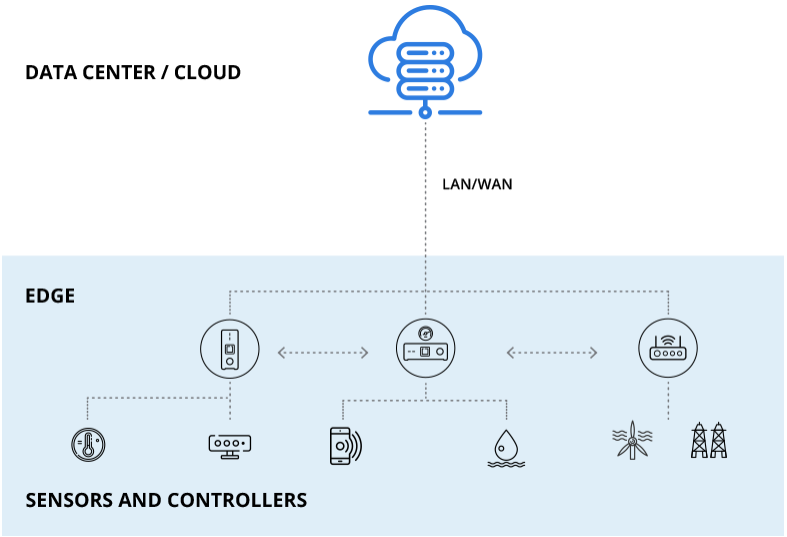
Edge Computing

Edge computing is an emerging computing paradigm which refers to a range of networks and devices at or near the user. Edge is about processing data closer to where it's being generated, enabling processing at greater speeds and volumes, leading to greater action-led results in real time.

Edge Comptuing

~~Big Challenges~~ >> Some solutions

- Data Security. Some IoT devices collect highly sensitive information.
- Data Privacy. Much of the information collected and processed by IoT devices may be protected under various data privacy laws. ...
- Data Volume. ...
- Data Complexity.



Contents

Edge Computing

1. Introduction
2. Requirements and Oportunities
3. Integrated Machine Learning

Contents

Embedded Systems & RTOS

Internet of Things

Edge Computing

After all – Goals

- Have some knowledge of previous topics
- Get in contact with microcontroller development, RTOS, IoT, ….
- Learn how to:
  - Allow multi-platform interaction
  - Overpass the initial hardware I/O limitations
  - To take benefit of RTOS on embedded controllers.
  - Collect and transfer data from remote sensors

Contents

Project Development

Health risk assessment - HRA

Health risk assessment - HRA

A health risk assessment (HRA), sometimes known as a health risk appraisal or health assessment, is a questionnaire that evaluates lifestyle factors and health risks of an individual. It collects health information, that can be typically coupled with a process that includes biometric testing to assess an individual's health status, risks, and habits.

Health risk assessment - HRA

A health risk assessment (HRA), sometimes known as a health risk appraisal or health assessment, is a questionnaire that evaluates lifestyle factors and health risks of an individual. It collects health information, that can be typically coupled with a process that includes **biometric testing** to assess an individual's health status, risks, and habits.

Biometric monitor framework

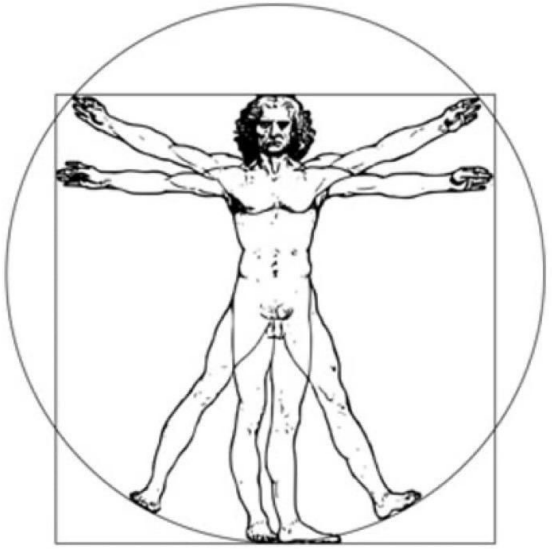
Project Statement:

- To develop a global system for monitoring user health risk situations based on some biometric data.
- The system will monitor:
  - The heart rate to determine possible heart arrhythmia or attacks.
  - The user movements and stability to identifies some possible neural problems due to a bad blood flow.
- The prototype is composed by several devices each one with different purposes:
  - Biometric data providers that obtains the biometric user data.
  - Monitor station: It shows in real time the current user status; heart rate and current risk.

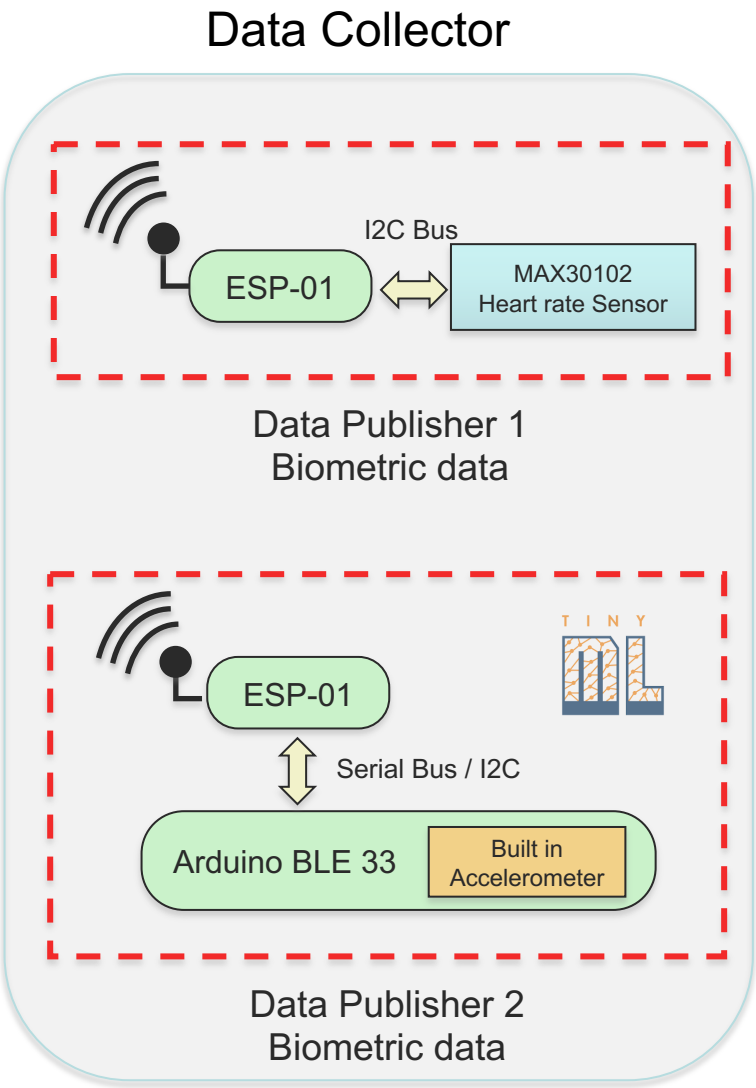
Biometric monitor framework

Project requirements:

- User wear two different devices that collect data; heart rate and movement/stability dysfunction.



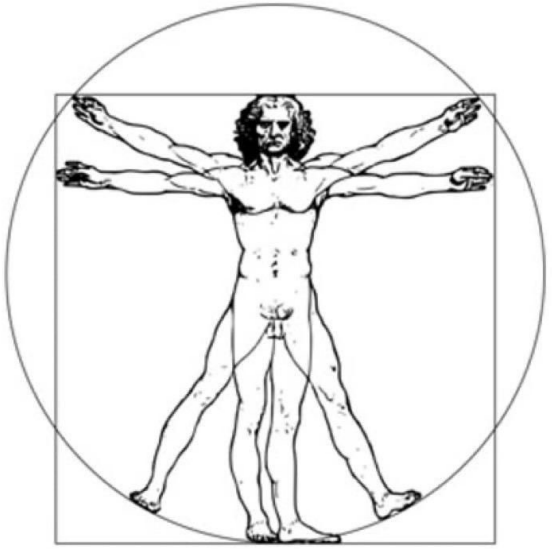
Biometric monitor framework



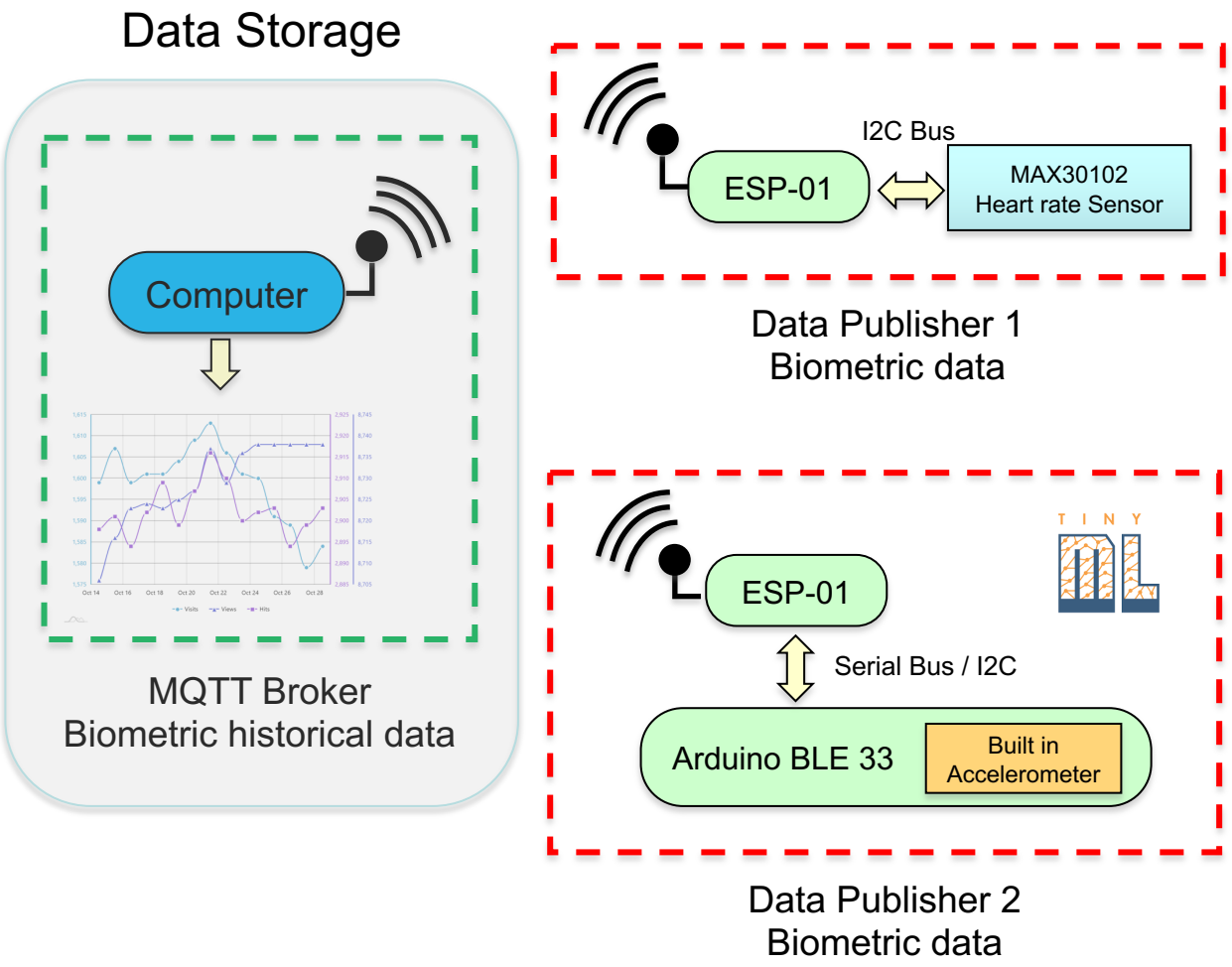
Biometric monitor framework

Project requirements:

- User wear two different devices that collect data; heart rate and movement/stability dysfunction.
- Data will be and transferred remotely.



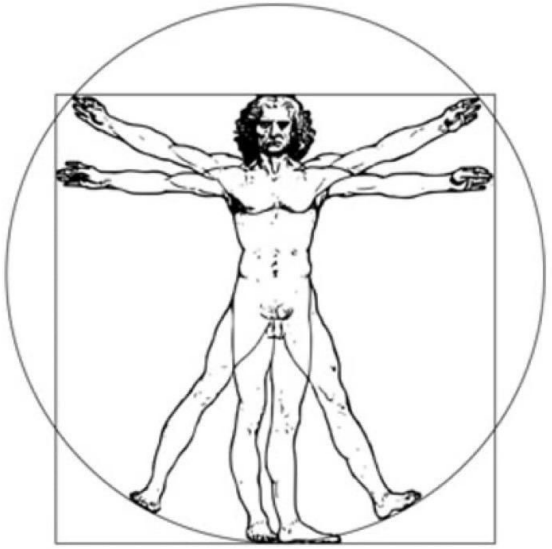
Biometric monitor framework



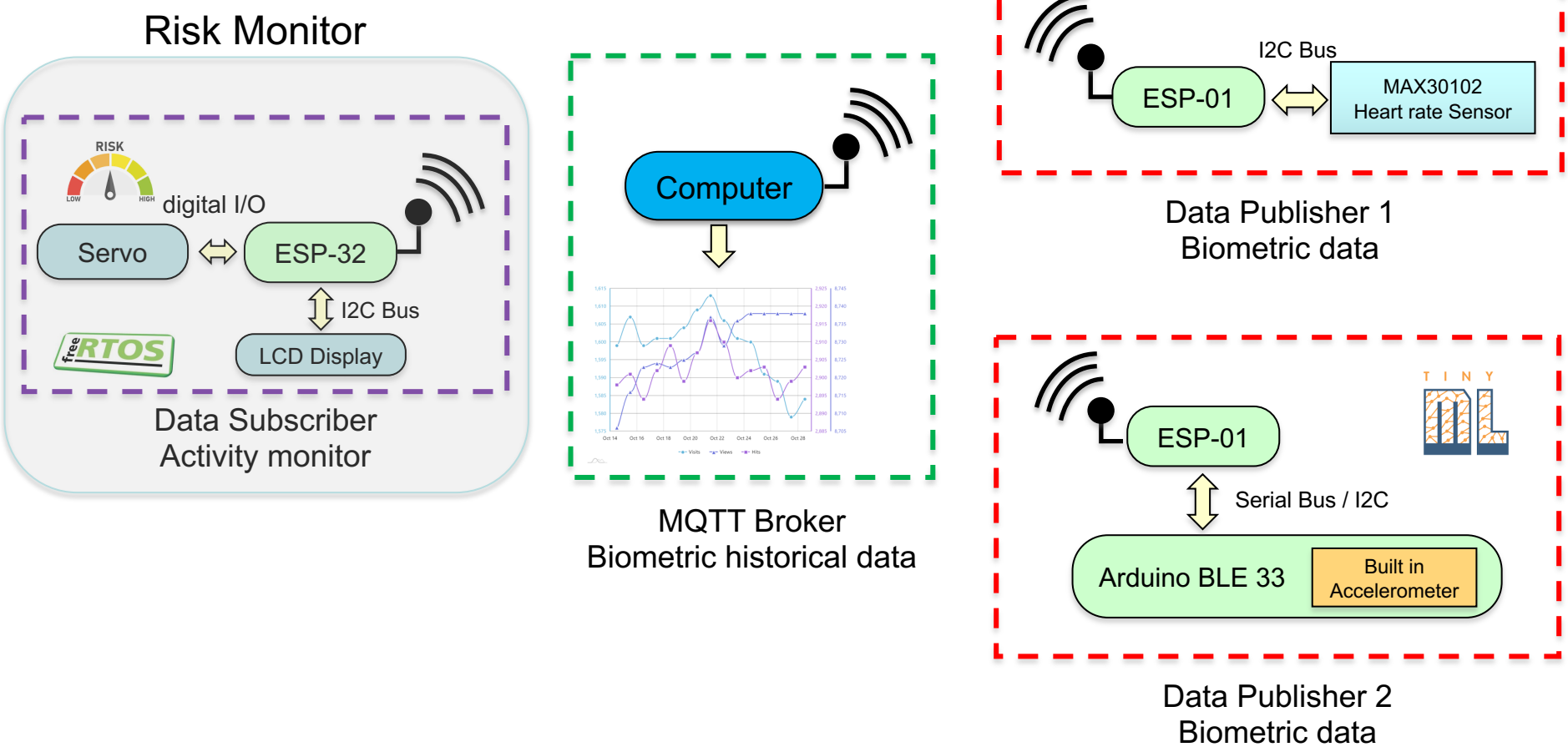
Biometric monitor framework

Project requirements:

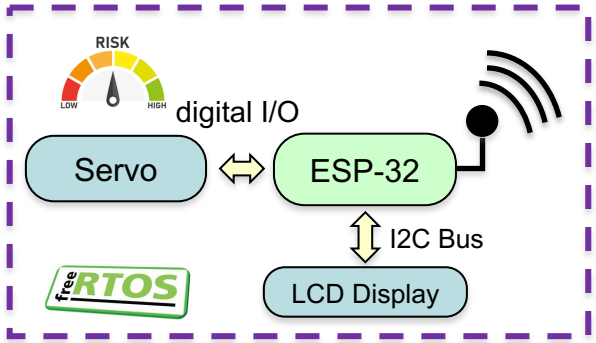
- User wear two different devices that collect data; heart rate and movement/stability dysfunction.
- Data will be and transferred remotely.
- There will be a risk monitor that identifies the current user status.



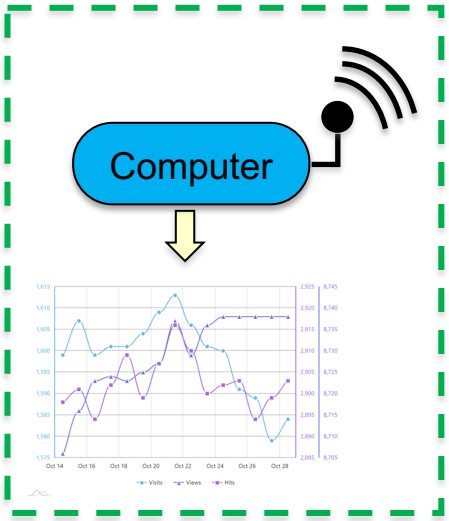
Biometric monitor framework



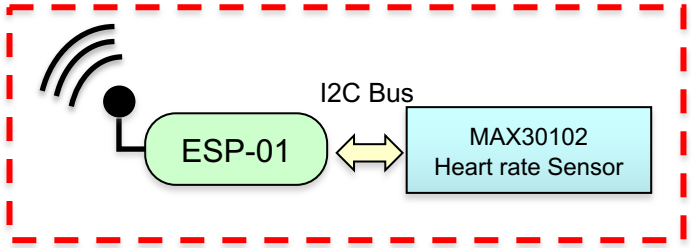
Prototype schema



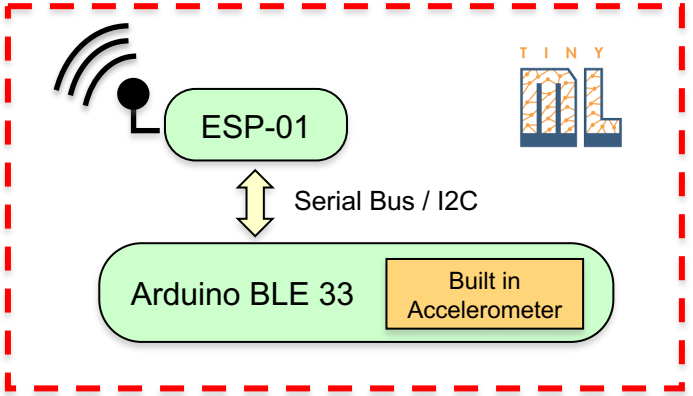
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



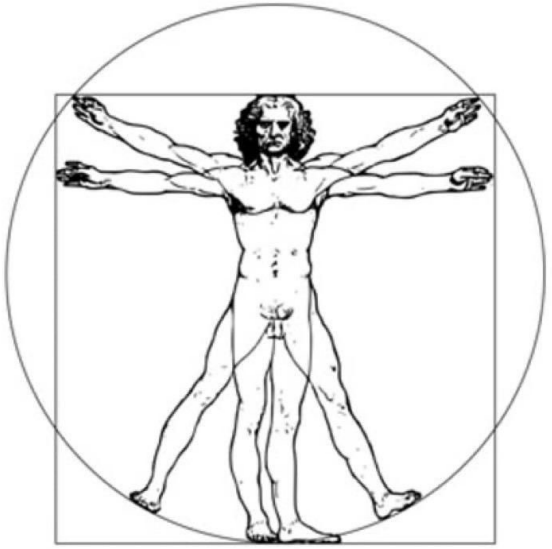
Data Publisher 1
Biometric data



Data Publisher 2
Biometric data

Biometric monitor framework

Project requirements:



- User wear two different devices that collect data; heart rate and movement/stability dysfunction.
- Data will be and transferred remotely.
- There will be a risk monitor that identifies the current user status.

Optional:

> Collected data can be remotely stored in a database.

> Smartphone data presentation.

> ….

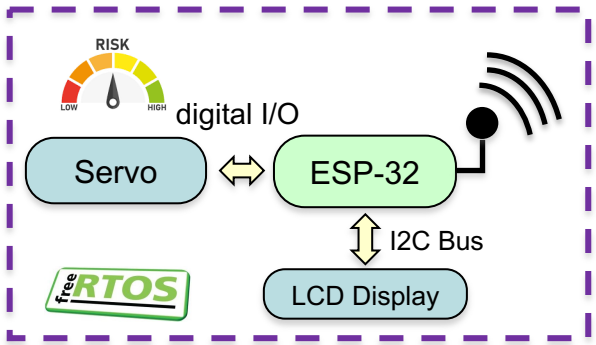
Project <> Topics

Embedded Systems & RTOS

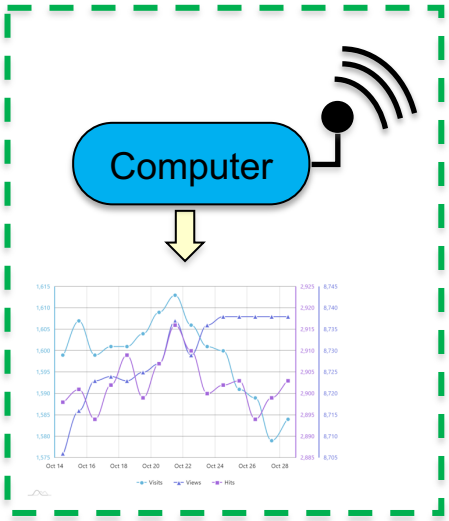
Internet of Things

Edge Computing

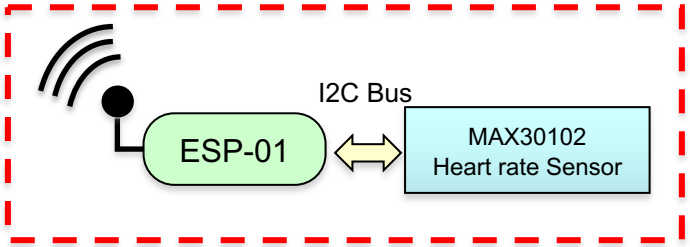
Prototype schema



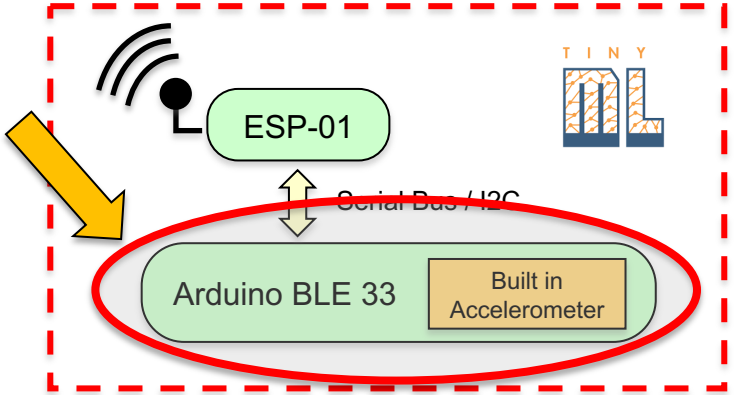
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data



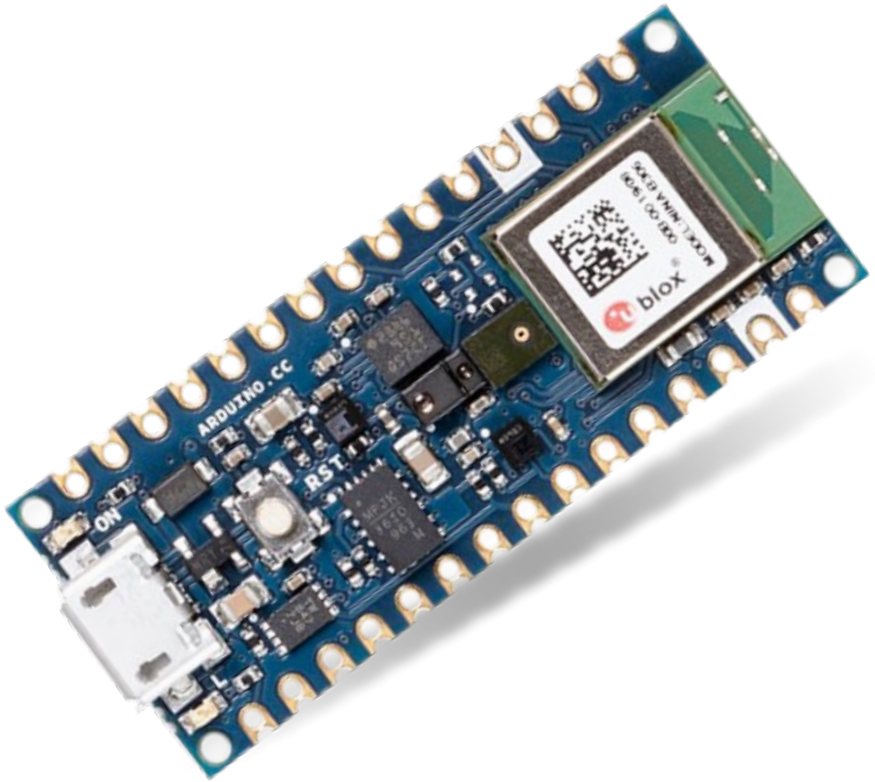
Data Publisher 2
Biometric data

Components - Arduino

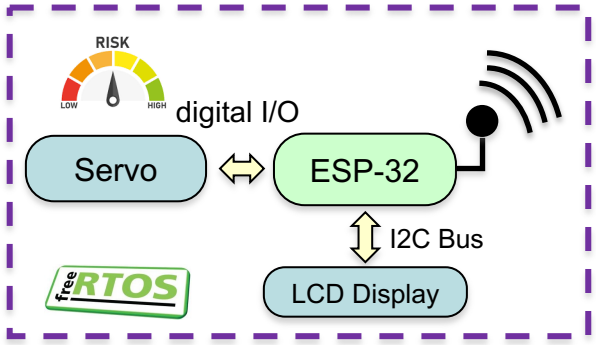
- Physical computing platform
- Several prototyping boards but just one IDE (many flavors)
- Open source
- “Hardware Abstracted” Wiring Language
- USB programmable
- Extremely large community
- Inexpensive (from 10$)

Components - Arduino

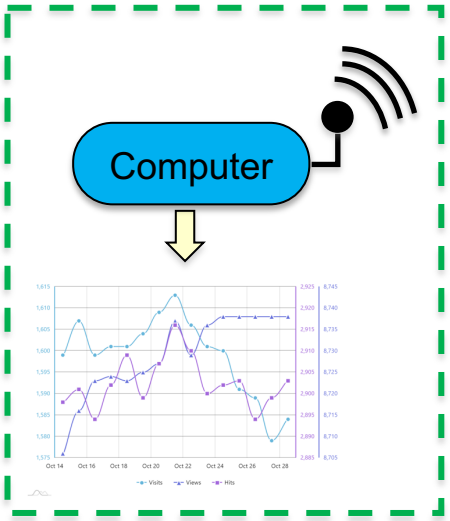
- nRF52840 - 32-bit ARM® Cortex®-M4 CPU – 64MHz
- 1MB RAM
- 34 × programmable GPIOs
- Bluetooth® Low Energy and Bluetooth® client and host device
- 1 x UART + 1 x SPI + 1 x I2C
- 8 ADC
- Embedded 9 axis inertial sensor



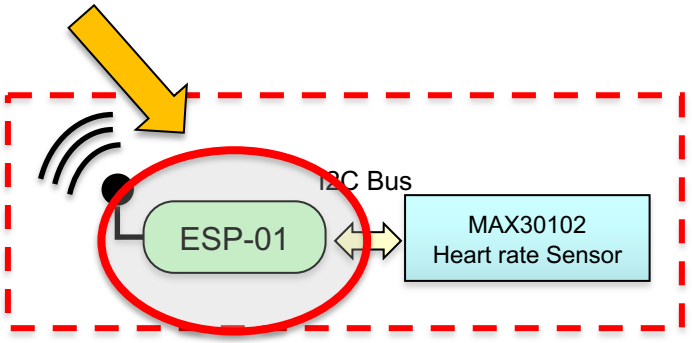
Prototype schema



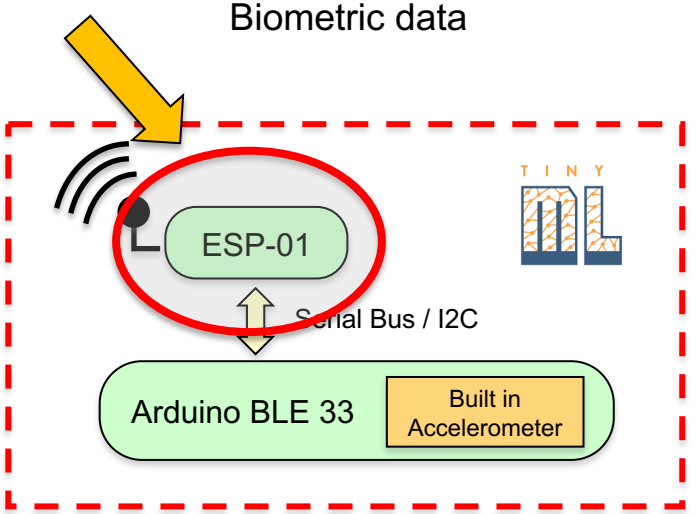
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data



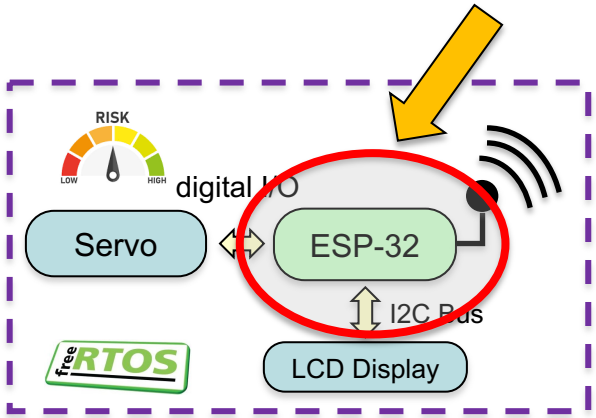
Data Publisher 2
Biometric data

Components – ESP-01

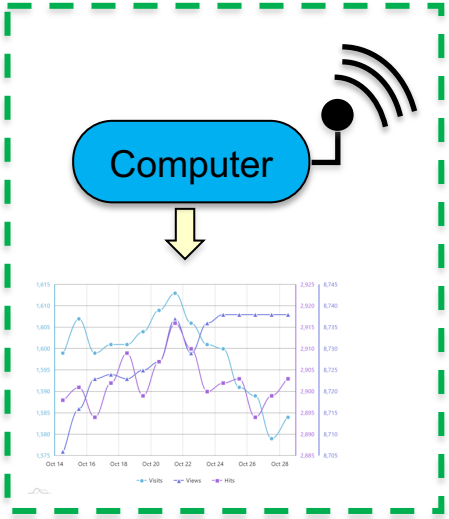
- Low cost, compact and powerful Wi-Fi Module
- Power Supply: +3.3V only
- Built-in low power ESP8266 32-bit MCU @ 80MHz
- 512kB Flash Memory
- Can be used as Station or Access Point or both combined
- Supports Deep sleep (<10uA)
- Supports serial communication hence compatible with many development platform like Arduino
- Can be programmed using Arduino IDE, AT-commands or Lua Scripting language
- UART + I2C Bus.

https://www.espressif.com/en/products/socs/esp8266
https://www.nodemcu.com/index_en.html

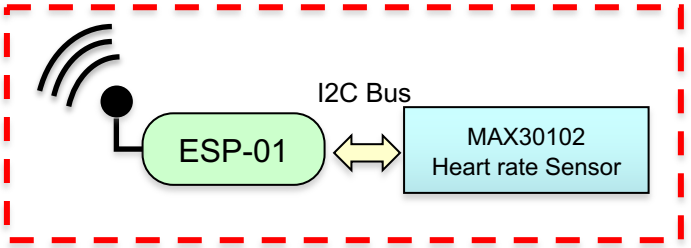
Prototype schema



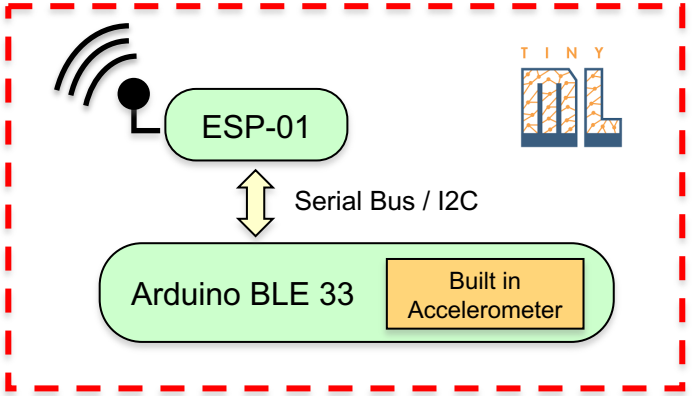
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data



Data Publisher 2
Biometric data

Components – ESP-32

- Power Supply: +3.3V only
- Xtensa dual-core 32-bit LX6 MCU @ 160MHz/240Mhz
- 320KB RAM + 512kB Flash Memory
- Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR and BLE
- 34 × programmable GPIOs
- 2 × 8-bit DACs
- 4 x SPI, 2 x I2C, 2 x I2S and 3 x Serial interfaces
- IEEE 802.11 standard security including WPA, WPA2, WPA3
- Cryptographic hardware acceleration
- ……

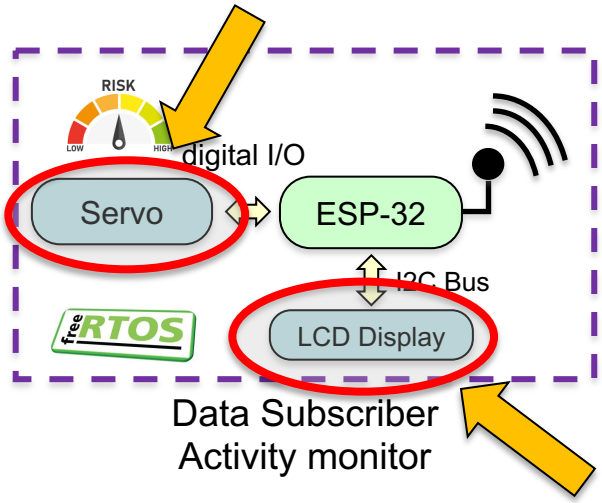
https://www.espressif.com/en/products/socs/esp32
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/

Components description

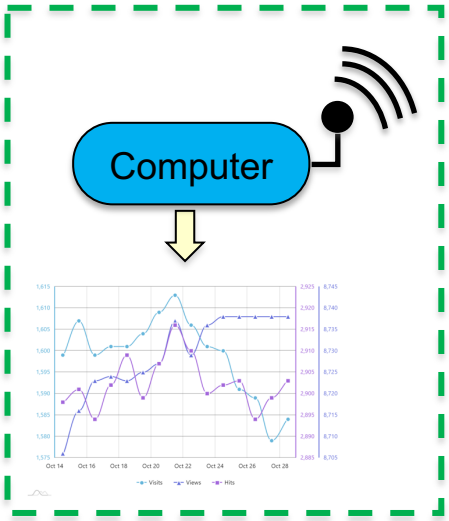
Sensors and Actuators

- Servo
- LCD Display
- Accelerometer
- Heart Rate

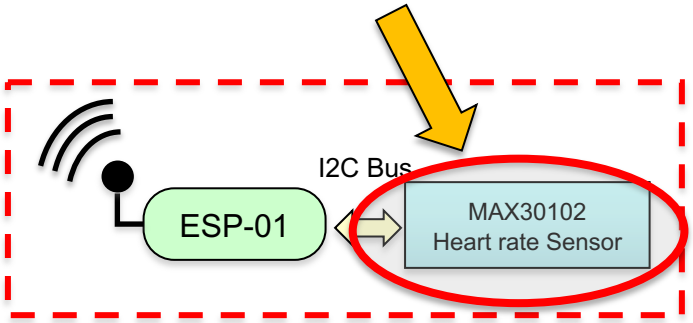
Prototype schema



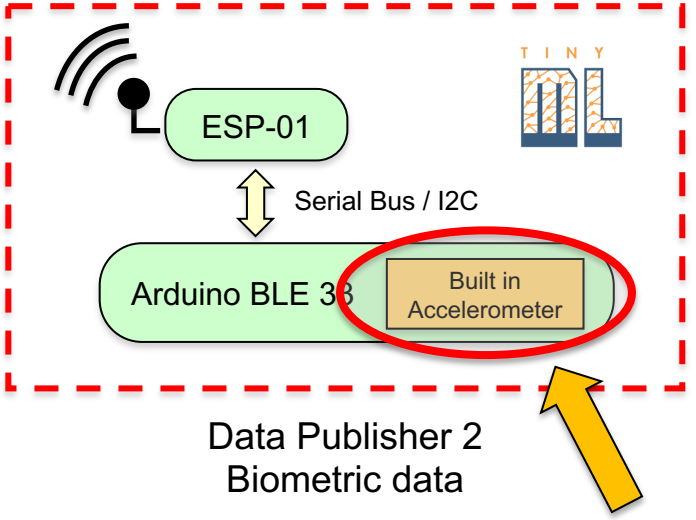
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data



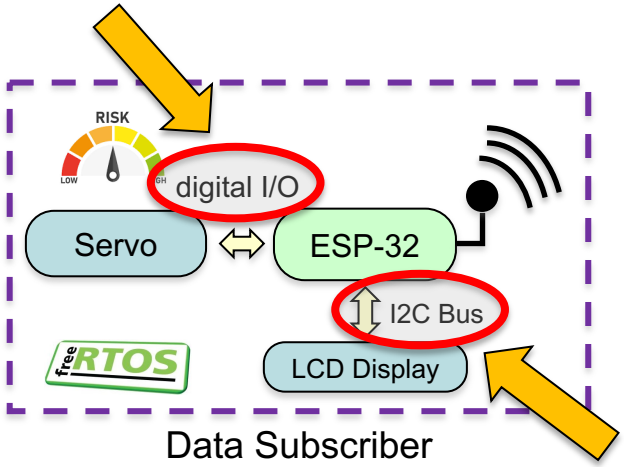
Data Publisher 2
Biometric data

Components description

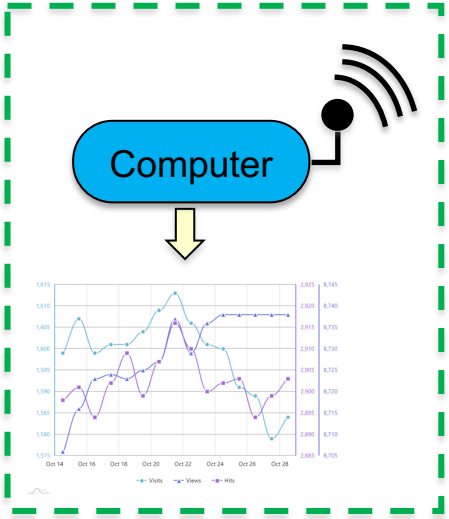
Communication Buses

- I2C – Device access
- Digital I/O – Device dependent

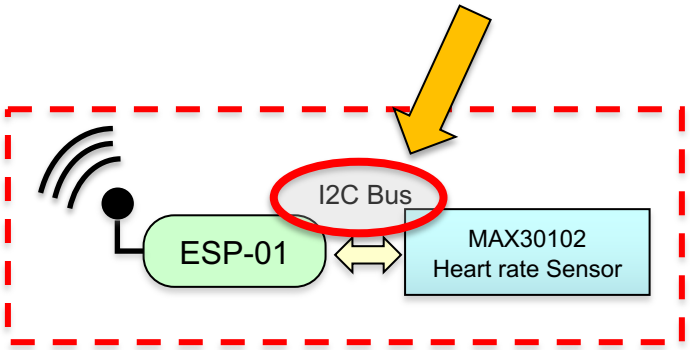
Prototype schema



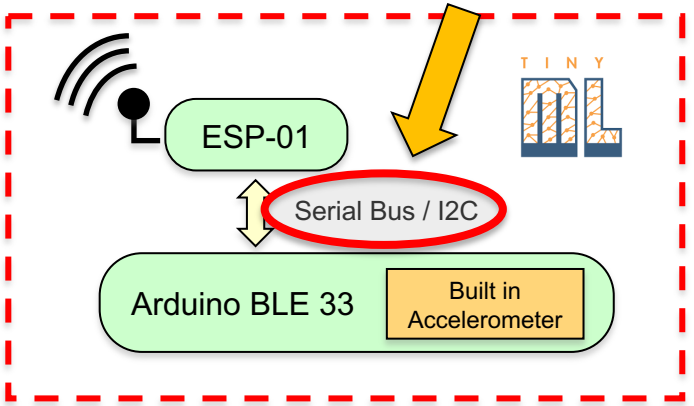
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data



Data Publisher 2
Biometric data

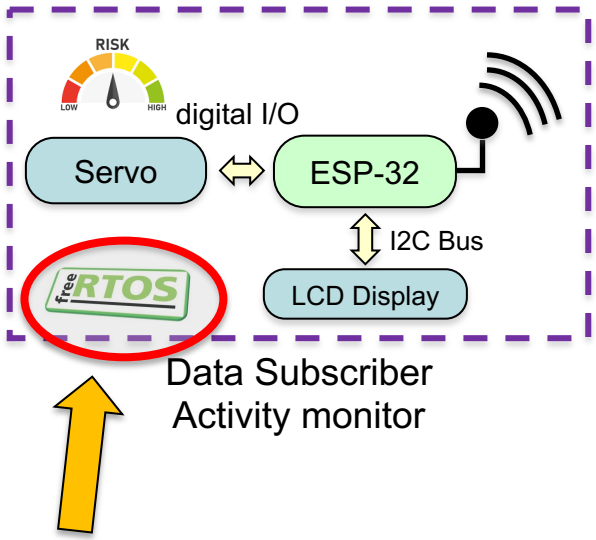
Project <> Topics

Embedded Systems **& RTOS**

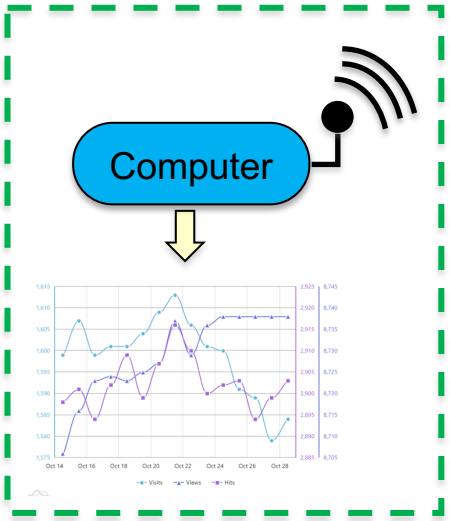
Internet of Things

Edge Computing

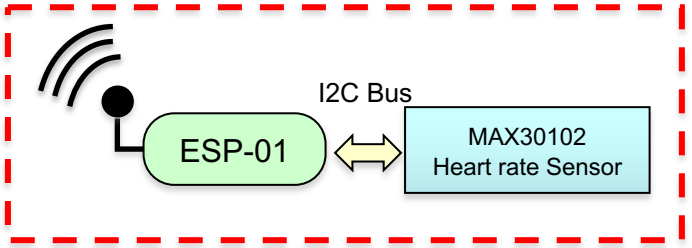
Prototype schema



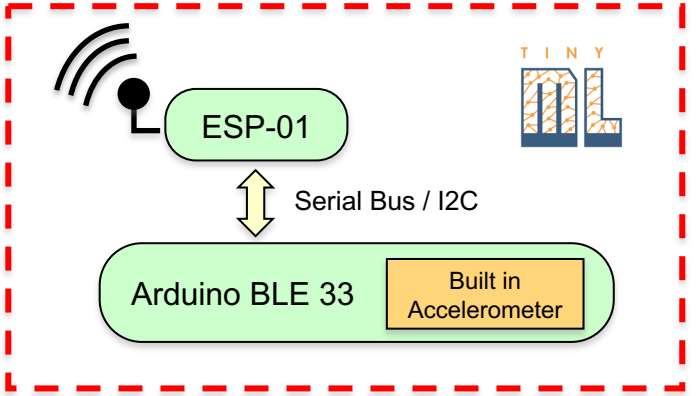
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



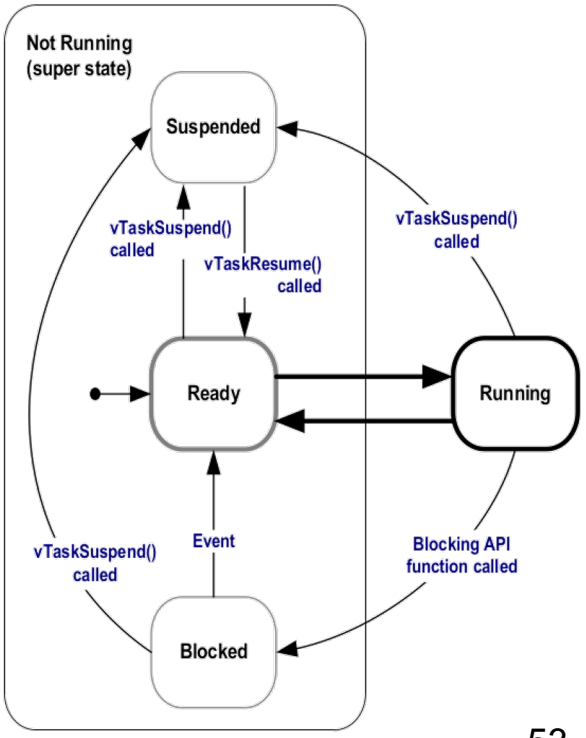
Data Publisher 1
Biometric data



Data Publisher 2
Biometric data

Components description

- Distributed freely under the MIT open-source license, > 25 architectures
- Minimal ROM, RAM and processing overhead
- The RTOS core kernel is contained in only 3 C files
- Allows MCU apps be organized as independent threads/tasks based on priorities.
- Pre-emptive or co-operative tasks
- Queues / Mutexes / Semaphores
- Tick / Idle hook functions



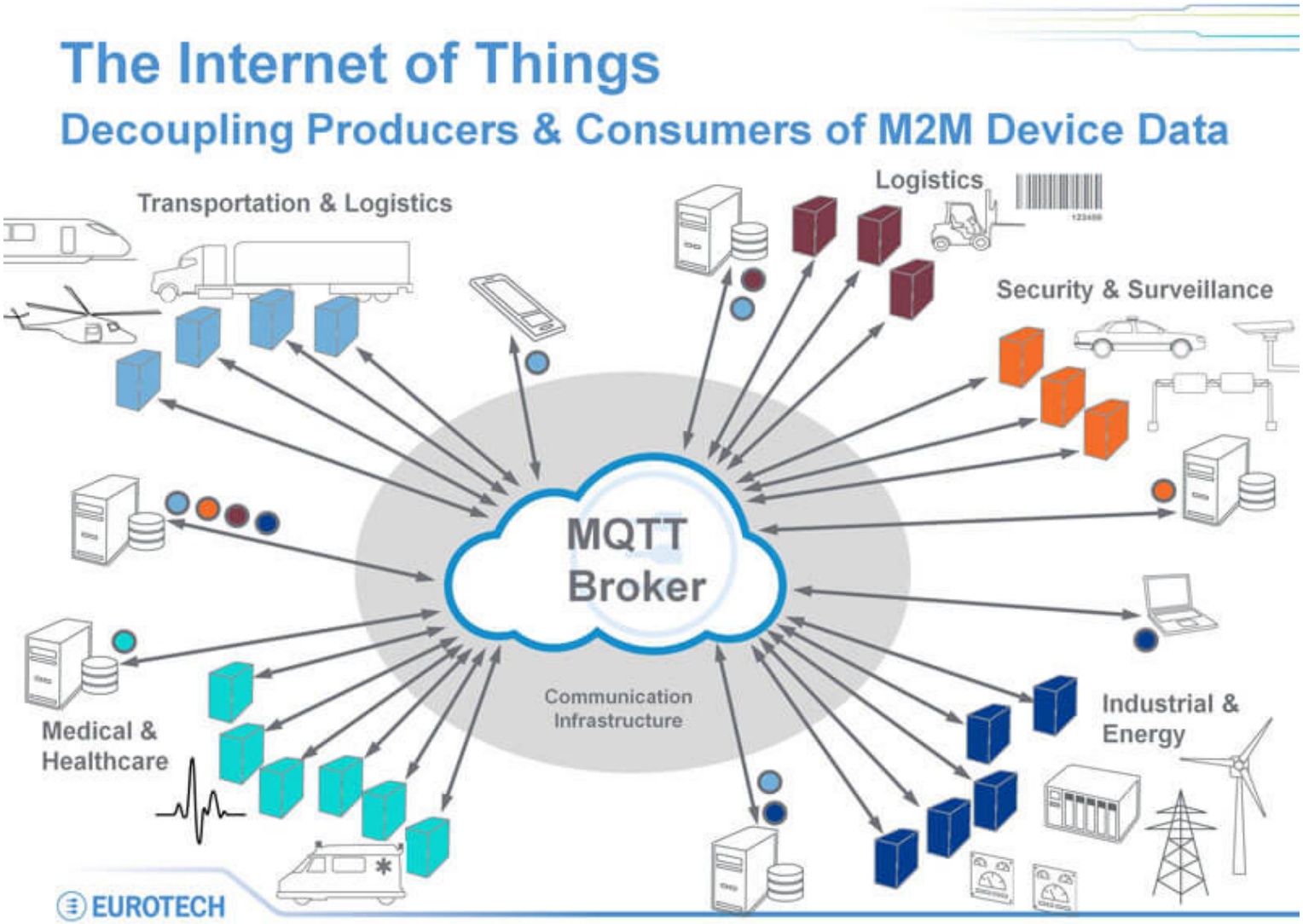
Project <> Topics

Embedded Systems & RTOS

Internet of Things

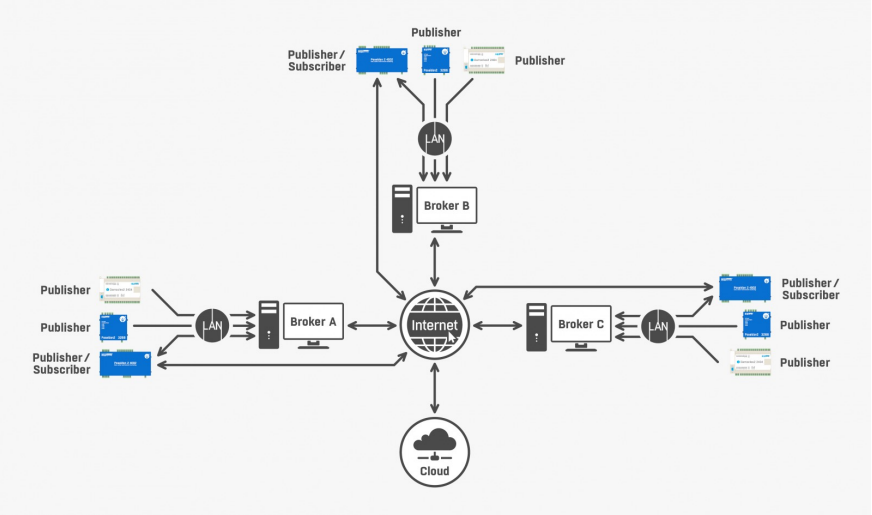
Edge Computing

IoT - MQTT


The Internet of Things
Decoupling Producers & Consumers of M2M Device Data
Transportation & Logistics
Logistics
Security & Surveillance
MQTT
Broker
Communication
Infrastructure
Medical &
Healthcare
Industrial &
Energy
EUROTECH

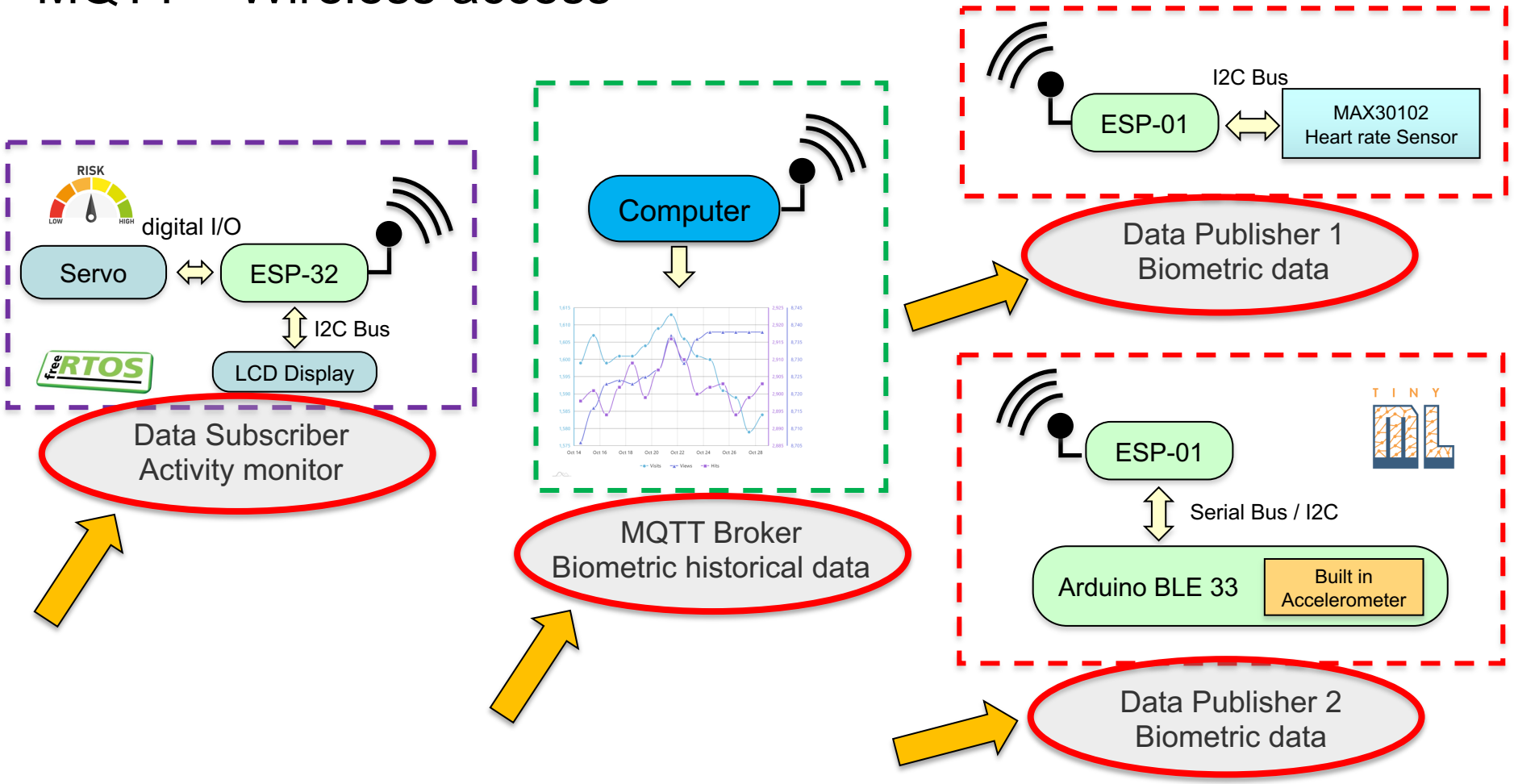
IoT - MQTT

- Client Server publish/subscribe messaging transport protocol.
- It is light weight, open, simple, and designed to be easy to implement.
- Independent of the payload content
- Three qualities of service for message delivery



Prototype schema

MQTT – Wireless access



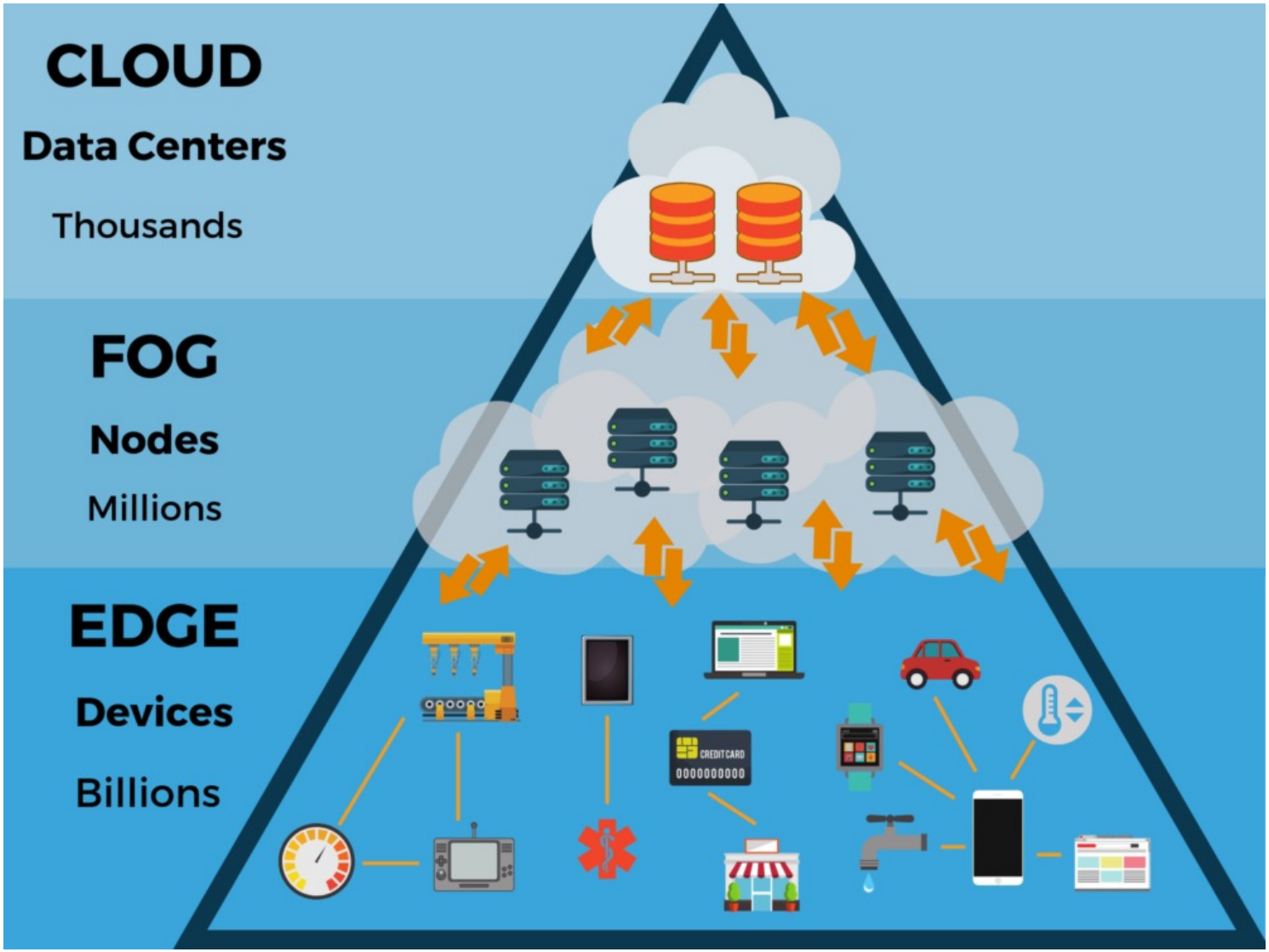
Project <> Topics

Embedded Systems & RTOS

Internet of Things

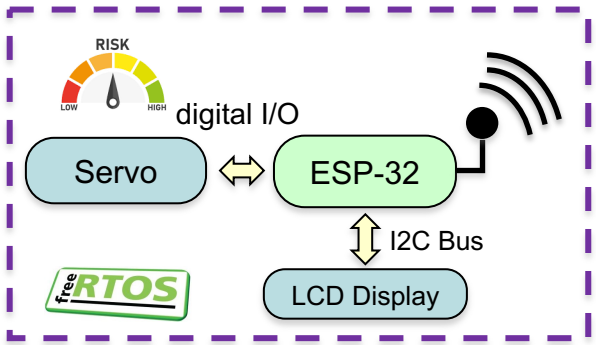
Edge Computing

Components description

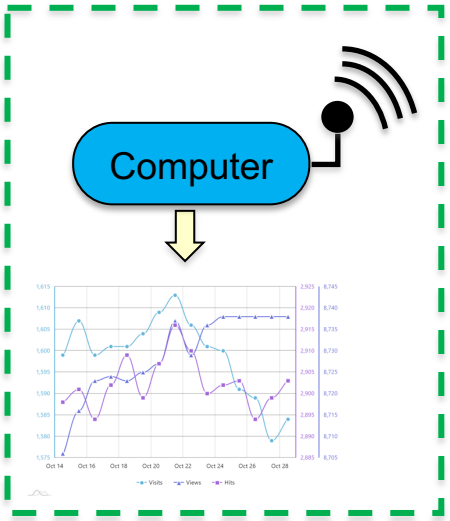


Prototype schema

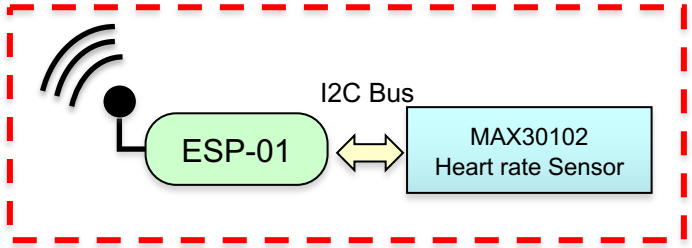
MQTT – Wireless access



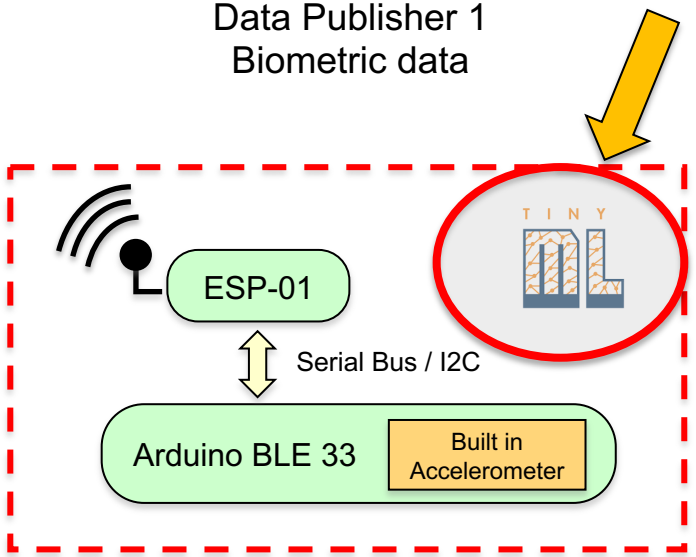
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data

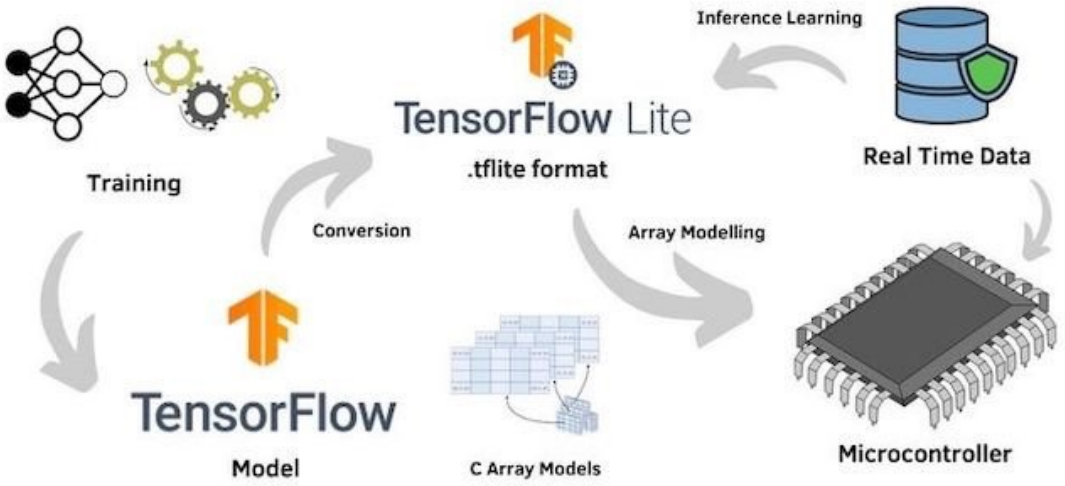


Data Publisher 2
Biometric data

Components description

Tiny-ML

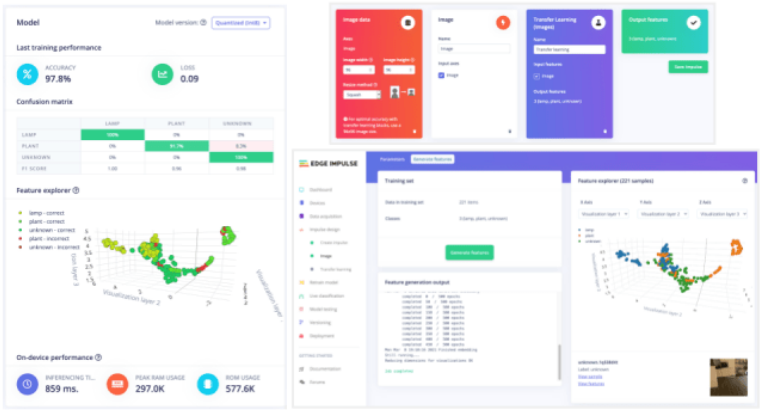
- Machine Learning applied to embedded systems.
- It utilizes neural networks to to recognize patterns.
- It is based on TensorFlow Lite

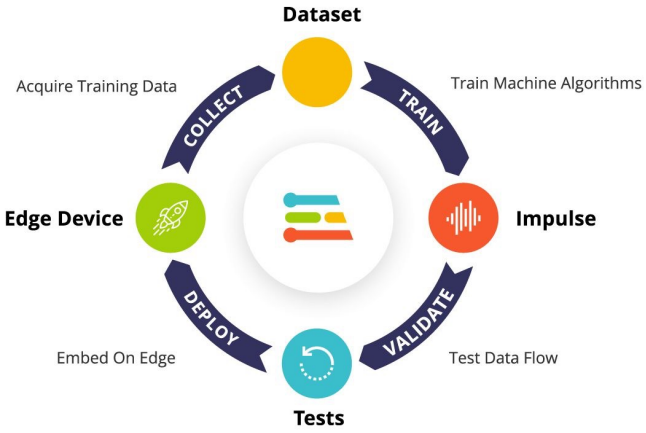


Components description

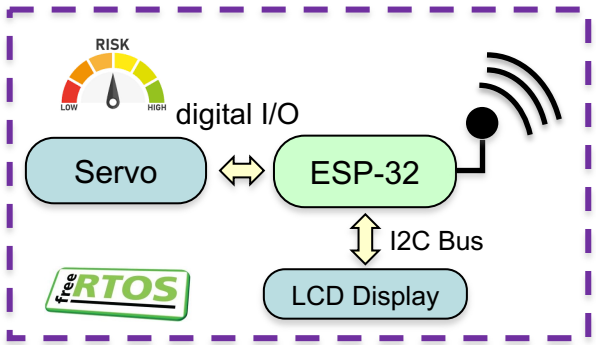
EDGE IMPULSE

- Company that provides the required tools to develop intelligent device solutions with embedded Machine Learning.
- It simplifies the NN development – Framework that makes the process of building, deploying, and scaling embedded ML applications easier and faster.

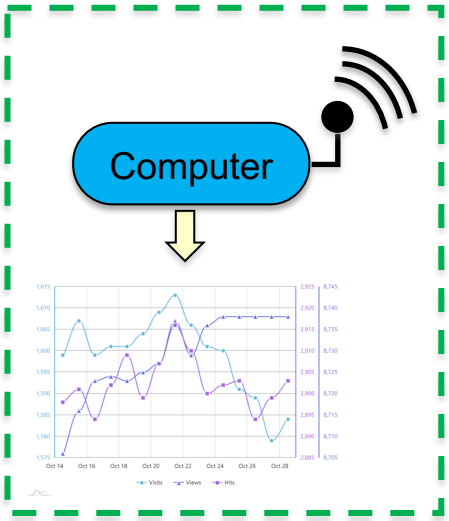




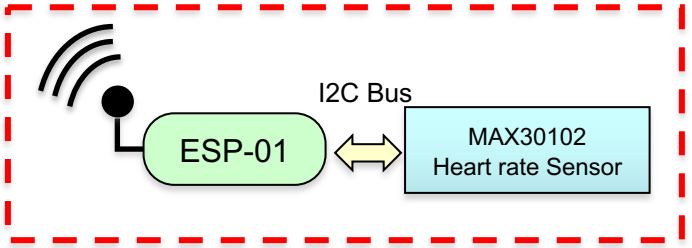
Prototype schema



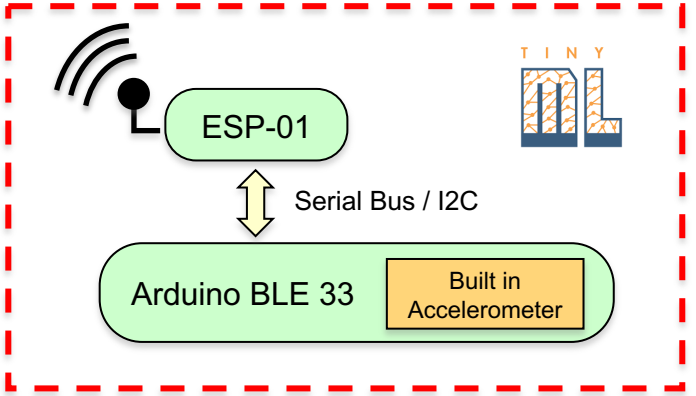
Data Subscriber
Activity monitor



MQTT Broker
Biometric historical data



Data Publisher 1
Biometric data



Data Publisher 2
Biometric data

Health risk assessment - HRA

Course structure

- Little theory presentation on the main topics
- Short exercises to get practice
- A development project based on SCRUM

Methodology

SCRUM – Agile teams

SCRUM – Agile Teams

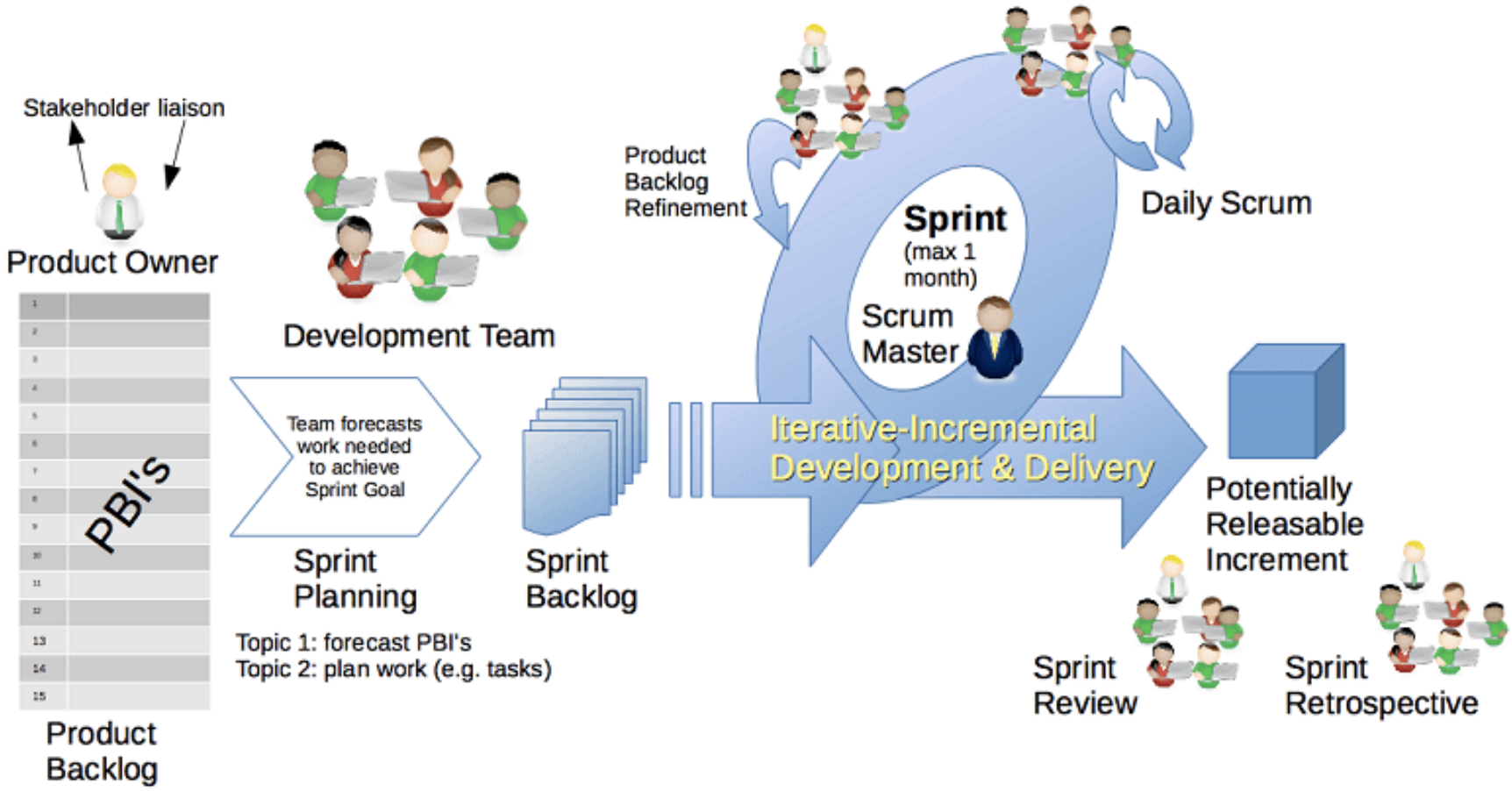
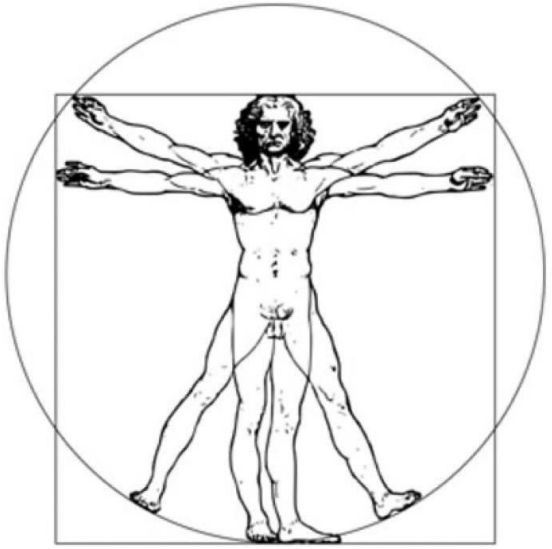


Image via Wikimedia Commons

SCRUM – Agile Teams

Project requirements:



- User wear two different devices that collect data; heart rate and movement/stability dysfunction.
- Data will be and transferred remotely.
- There will be a risk monitor that identifies the current user status.

Optional:

> Collected data can be remotely stored in a database.

> Smartphone data presentation.

> ….

User stories

1. It is required to obtain the user heart rate
2. It is necessary to determine the user movement and determine any stability dysfunction
3. MQTT protocol will be adopted to transfer and share the data
4. The monitor risk must show in real time the current user status

SCRUM – Agile Teams

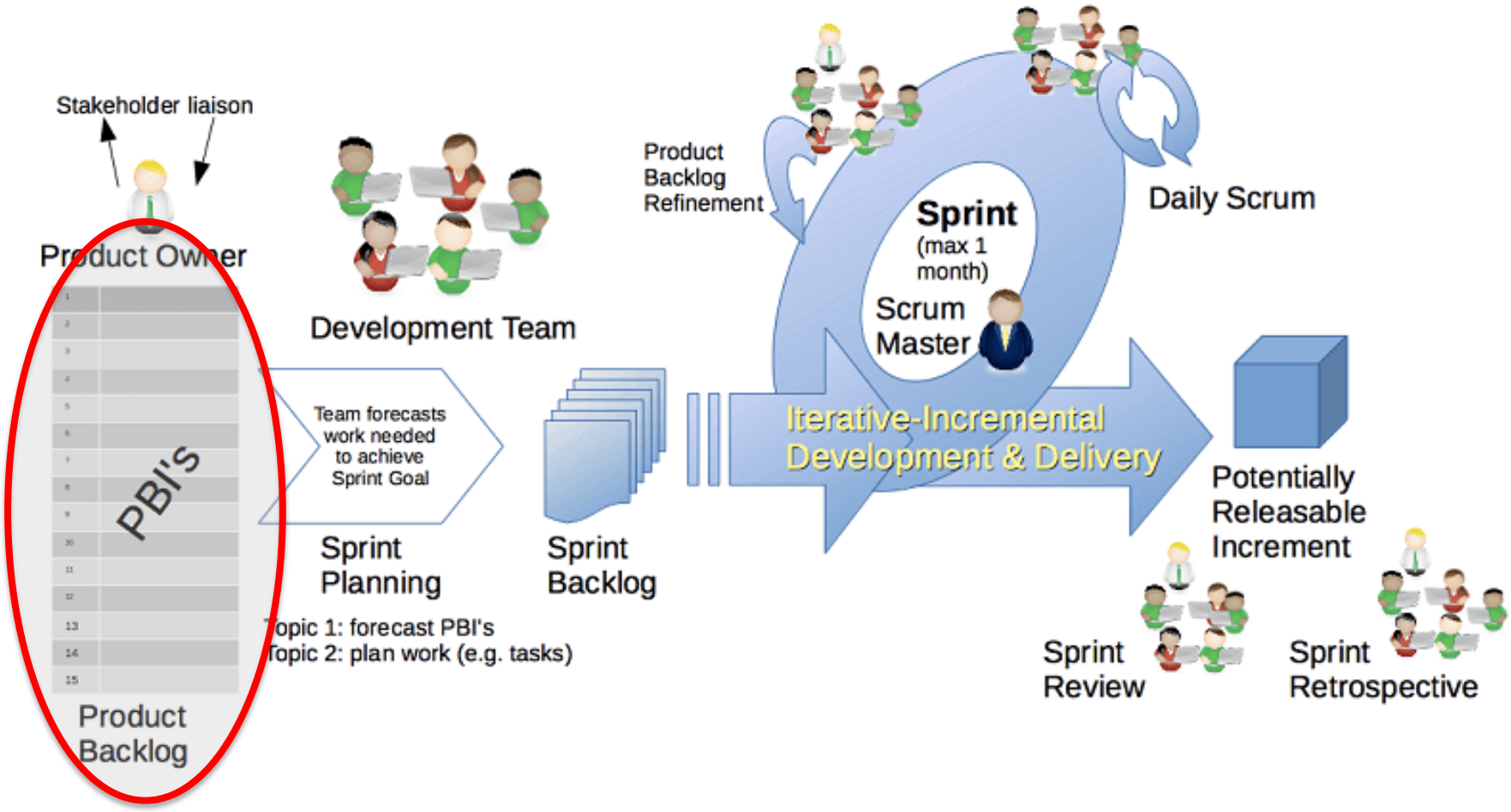


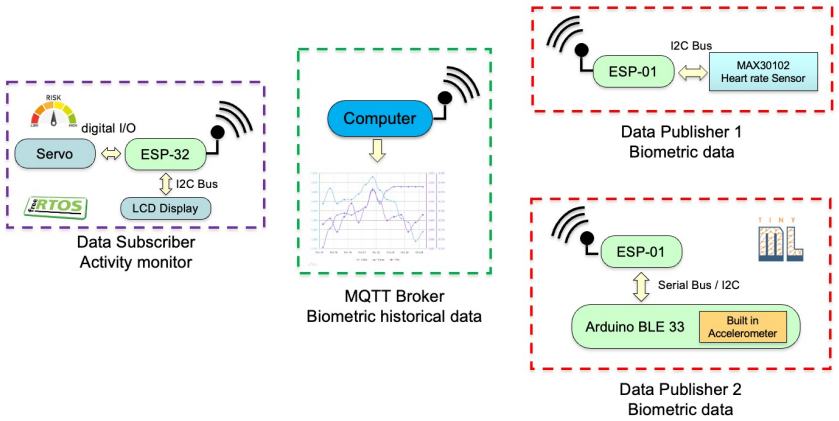
Image via Wikimedia Commons

SCRUM – Agile Teams

User stories

1. It is required to obtain the user heart rate
2. It is necessary to determine the user movement and determine any stability dysfunction
3. MQTT protocol will be adopted to transfer and share the data
4. The monitor risk must show in real time the current user status

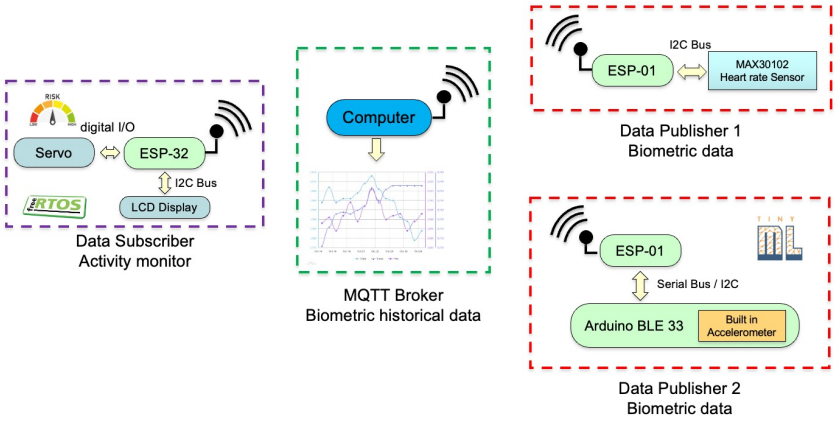
Identifying simple PBIs



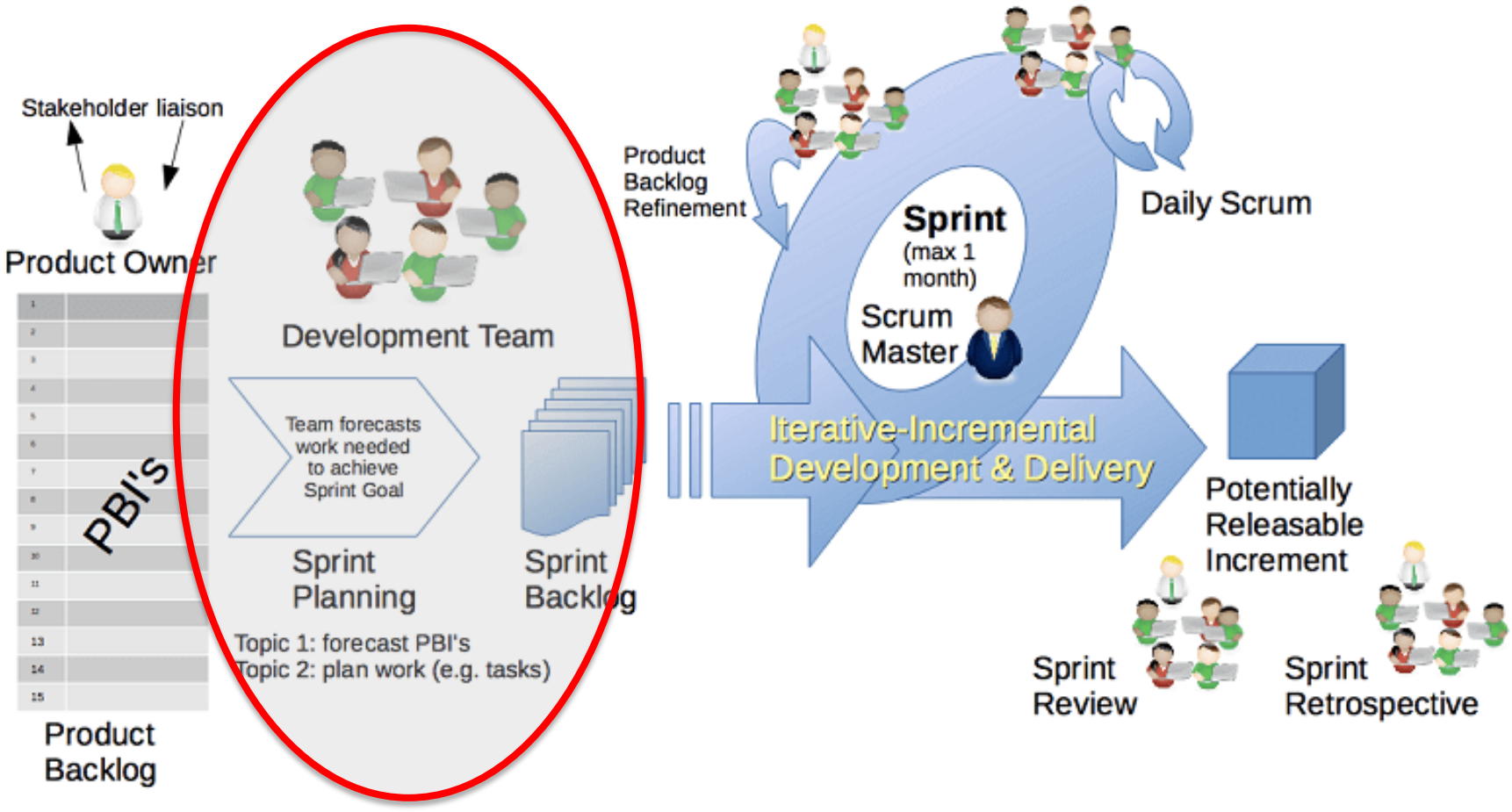
SCRUM – Agile Teams

Product Backlog Items:

- Prepare the IDE - Arduino, ESP-01, ESP-32
- Define the test&check process
- How to → I2C protocol
- How to → MAX30102
- How to → LCD Display
- ESP-01 ←→ Arduino interaction
- Data Producer 1 development
- Data Producer 2 development
- Data Producer 2 → Accelerometer Machine Learning
- MQTT Broker and Publish and Subscription process.
- Supervision Station: FreeRTOS implementation



SCRUM – Agile Teams



Identify groups of PBI and define the sprints

Image via Wikimedia Commons

Scheduling & Sprints

- Each group should define its own sprint planning and roadmap.
- Lectures will be focused on developing, solving doubts and if required to explain some content.

SCRUM – Agile Teams

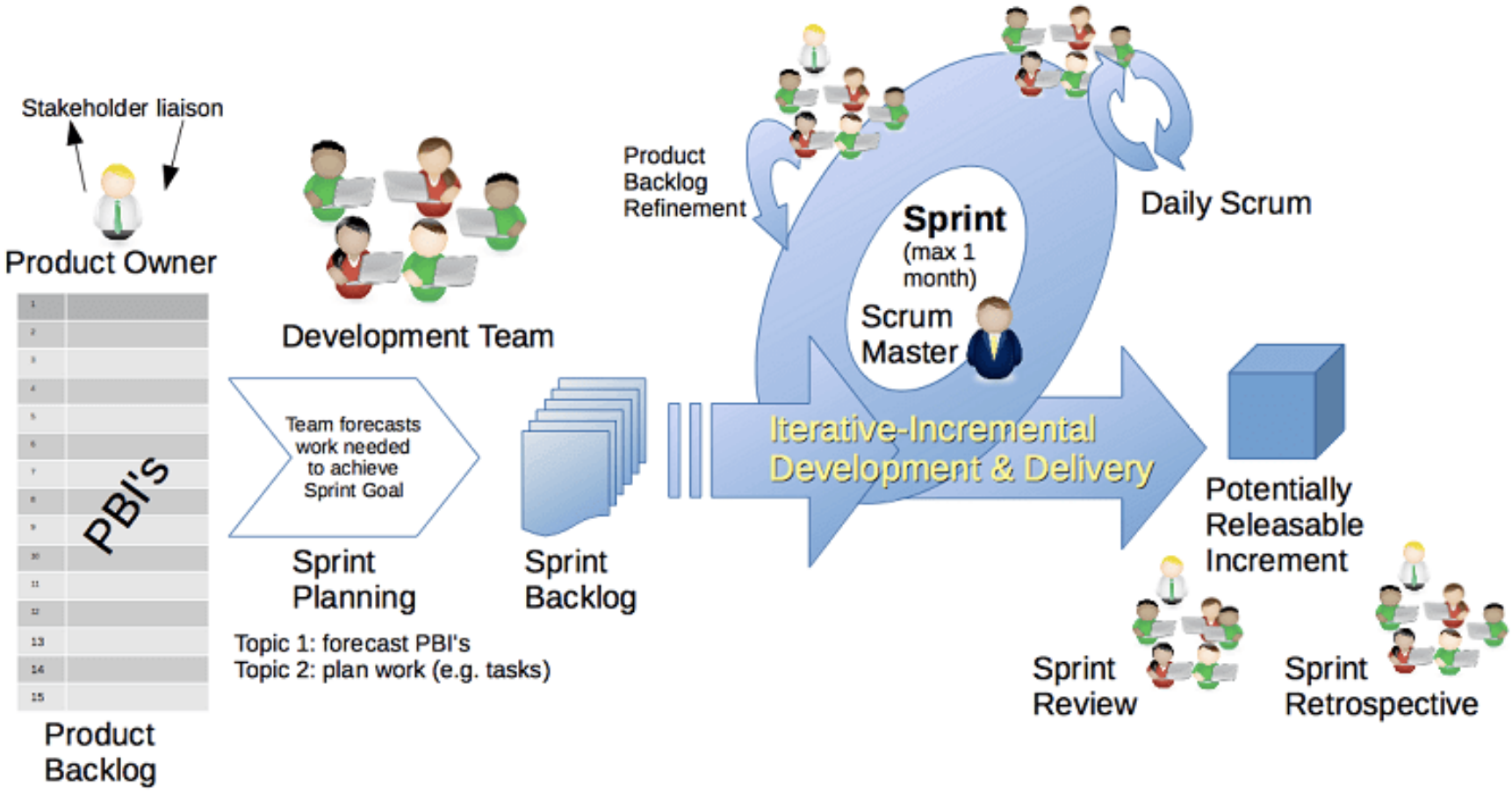


Image via Wikimedia Commons

SCRUM – Agile Teams

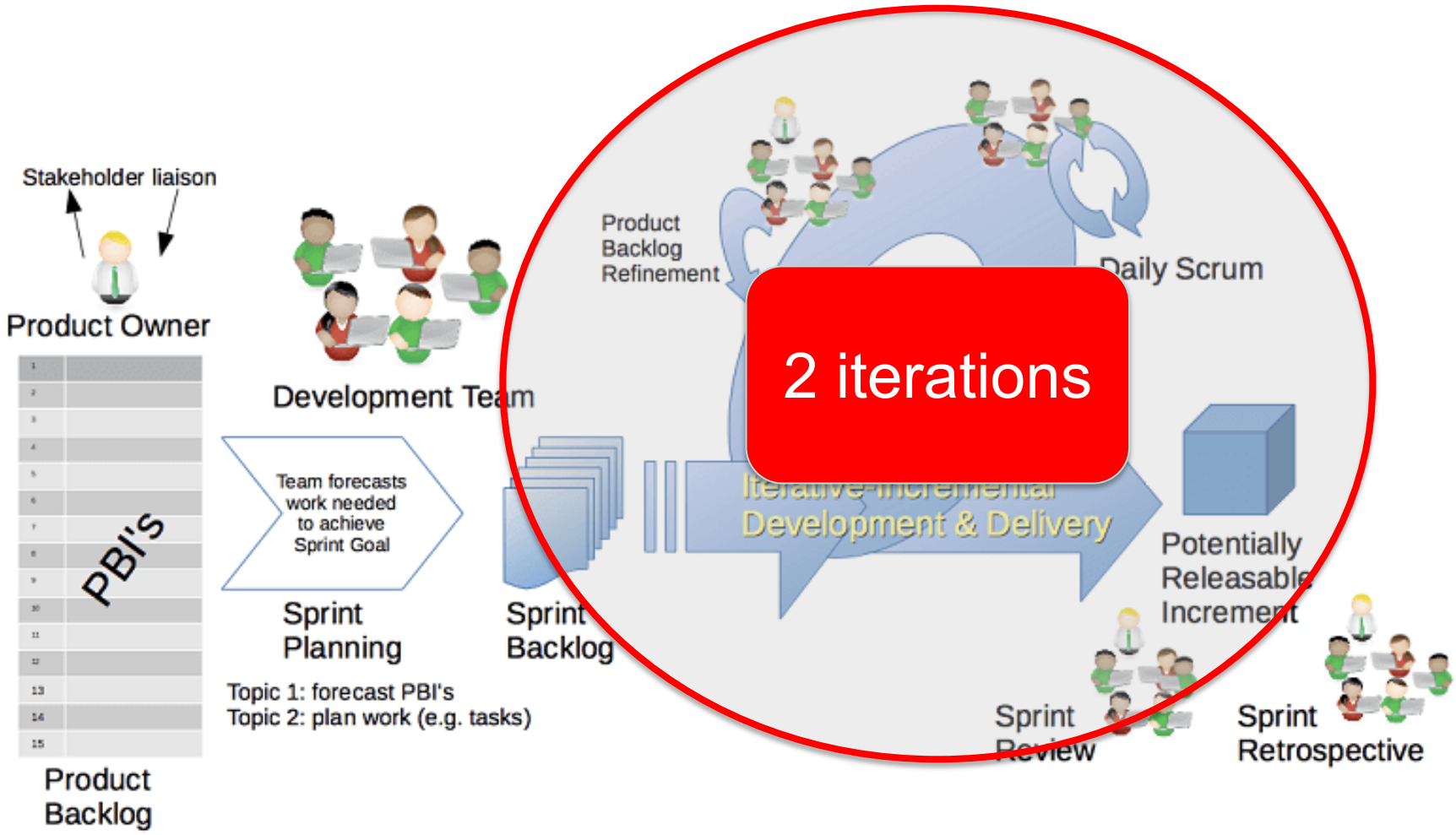


Image via Wikimedia Commons

SCRUM – Agile Teams

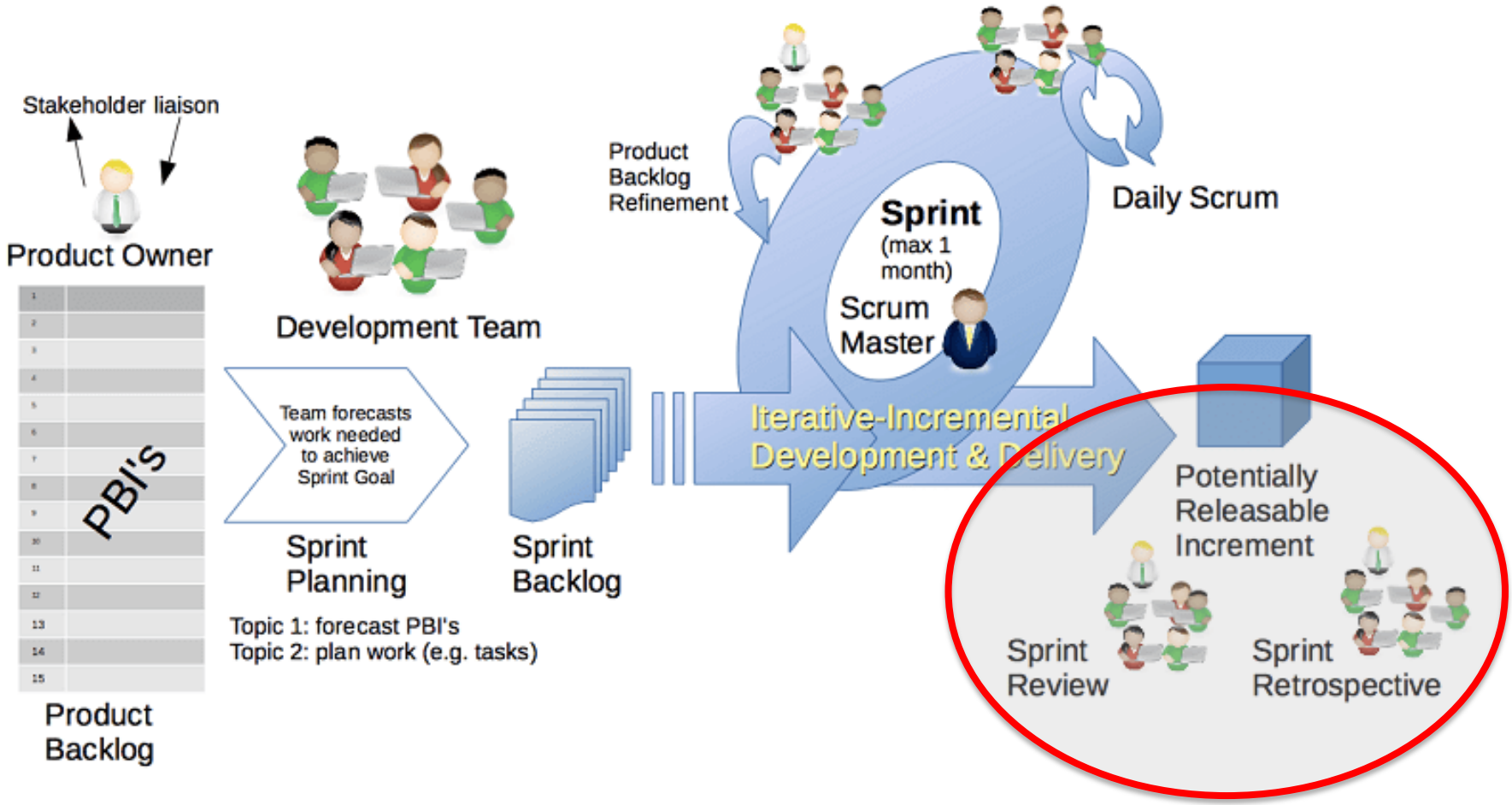


Image via Wikimedia Commons

Scheduling & Sprints

- Each group should define its own sprint planning and roadmap.
- Lectures will be focused on developing, solving doubts and if required to explain some content.
- Sprint Sessions:
  - 1st part – Each team ➔ Group Sprint delivery
    - Tasks completion and defence
    - Identify next task to continue

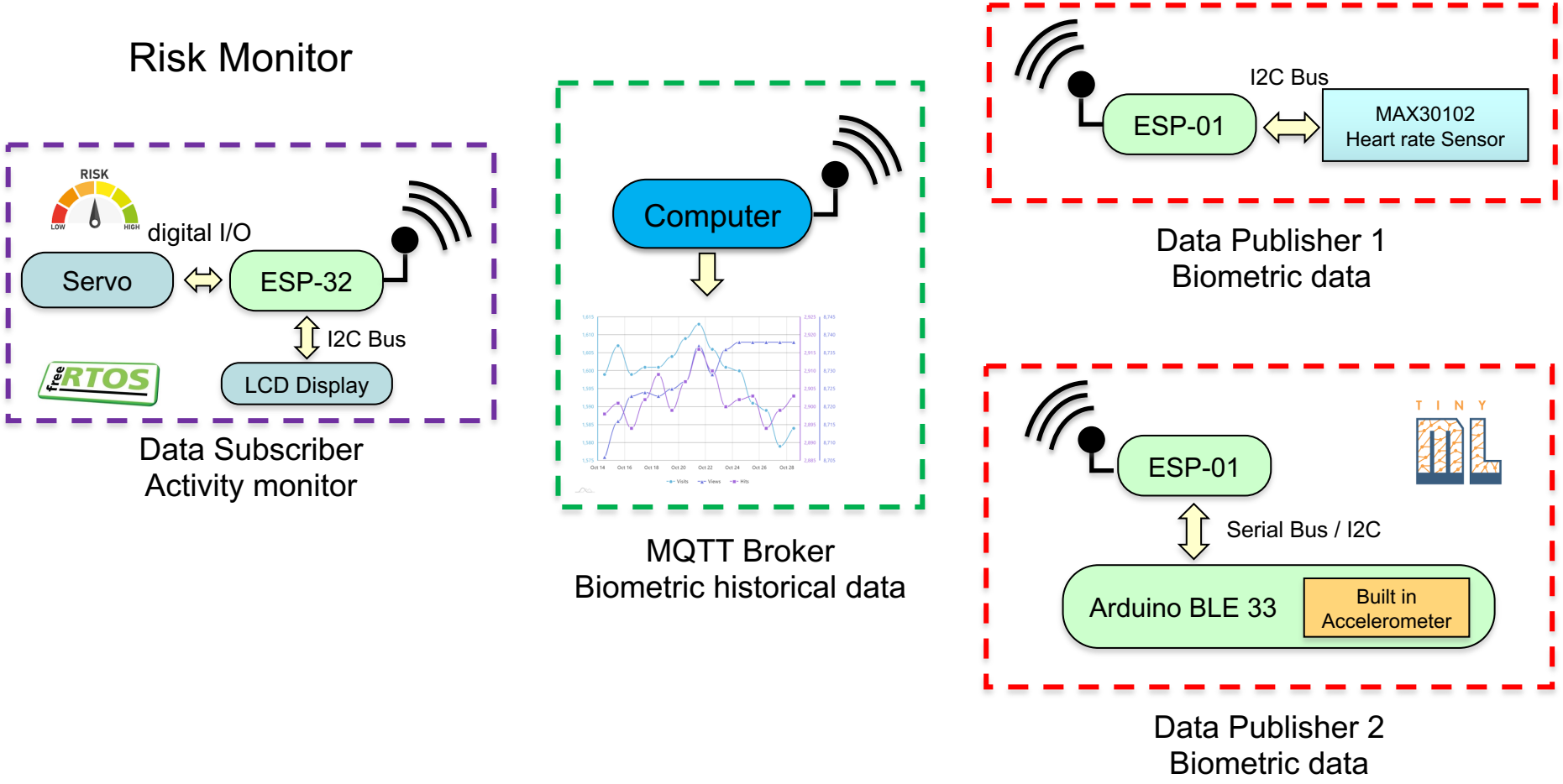
Scheduling & Sprints

- Each group should define its own scheduling and roadmap.
- Lectures will be focused on developing, solving doubts and if required to explain some content.
- Sprint Sessions:
  - 1st part – Each team ➔ Group Sprint delivery
    - Tasks completion and defence
    - Identify next task to continue
  - 2nd part – All teams together
    - Each team ➔ Development status presentation
    - Global Sprint Review and Retrospective

Team – Starting Point

- For each PBI define the corresponding story points assignment → Identify the required effort and grade
- Determine the first sprint
- Velocity estimation vs current progression

Biometric monitor framework



Content

Course Scheduling

Course Schedule

septiembre

L	M	X	J	V	S	D
			1	2	**3**	**4**
5	6	7	8	9	**10**	**11**
12	13	14	15	16	**17**	**18**
19	20	21	22	23	**24**	**25**
26	27	28	29	30		

octubre

L	M	X	J	V	S	D
					1	**2**
3	4	5	6	7	**8**	**9**
10	11	12	13	14	**15**	**16**
17	18	19	20	21	**22**	**23**
24	25	26	27	28	**29**	**30**
31						

noviembre

L	M	X	J	V	S	D
	1	2	3	4	**5**	**6**
7	8	9	10	11	**12**	**13**
14	15	16	17	18	**19**	**20**
21	22	23	24	25	**26**	**27**
28	29	30				

diciembre

L	M	X	J	V	S	D
			1	2	**3**	**4**
5	6	7	8	9	**10**	**11**
12	13	14	15	16	**17**	**18**
19	20	21	22	23	**24**	**25**
26	27	28	29	30	**31**	

Computer Graphics

Embedded Systems

Intelligent Systems

Course Schedule

23th September
Introduction to the subject
Project Presentation

07th October
> Embedded Systems → Topic
> IDE Development

14th October
> Embedded Systems → I2C Bus

Development, Questions, Product Backlog Refinement

20th October
> Embedded Systems – ESP01, ESP32
> IoT - MQTT Message Protocol

21th October
1st Sprint
Sprint Delivery, Review and Retrospective

> ESP01 & ESP32 programming

04th November
> RTOS -> FreeRTOS / ESP32

11th November
> Edge Computing
> Embedded Machine Learning – Arduino Nano BLE 33

18th November
2nd Sprint
Sprint Delivery , Sprint Review and Retrospective

19th January - Project Defense

Content

Grade

Grade

Activity	% grade	Minimum grade	Group/Individual	Mandatory
Peer evaluation (1)	30%	No	Group	Yes
Self evaluation	10%	No	Group and individual	Yes
Professor evaluatión	10%	No	Group	Yes
Project documentation	10%	No	Group	Yes
Project development and results	25%	No	Group and individual	Yes
Project defense (1)	15%	No	Individual	Yes

In each Sprint there will be professor, class group, team and self evaluation.

(1) Implies an oral presentation

Grade

Scheduling & Sprints ➔ GRADE

- Sprint Sessions:

Sprint Documentation > Items evaluated: Documentation (10%)

1. Technical document
    a. Backlog tasks.
    b. Dedication time to each task (an estimation in hours for the whole team).
    c. Sprint definition – A short self-explaining sentence related to the user story that identify the sprint goal.
    d. Percentage of Sprint completion (or individual tasks).
    e. Sprint Review – A short conclusion on the present Sprint.

Grade

Scheduling & Sprints → GRADE

- Sprint Sessions:

Scrum Master > Items evaluated: Professor (10%) + Development&Results (25%)

2. Development status

   The scrum master could make questions and suggestions at the end of the presentation.
   a. If required Source code + associated documentation (preferable linked to Github)
   b. If required a wiring schematic that represents the hardware module.
   c. Oral presentation. (Remember to **clearly** identify each member contribution)

3. Sprint retrospective

   a. Anything you think it must be redefined from the backlog → it will be evaluated and agreed by the product owner.
   b. Backlog Tasks grooming (if required).

4. Next Sprint definition

   a. Selected Backlog tasks.
   b. Sprint definition – A short self-explaining sentence related to the user story.

Scheduling & Sprints ➔ GRADE

- Sprint Sessions:

Sprint Class Presentation > Items evaluated: Peer evaluation (30%) + Self-evaluation (10%)

5. Class-evaluation.

   The team will present to the class-mates:
   a. The User story.
   b. The Sprint definition.
   c. The proposed solution and result.

   At the end of the presentation, any other student can make questions.
   The teams will evaluate each presentation, and they will mark each student. Additionally, each team can assign by consensus an extra point to one member of each other team.
   Remember to **clearly** identify each member contribution and try to fair distribute the presentation to facilitate the class-evaluation.

6. Auto-evaluation.

   Each team will fulfil two external questionnaires:

   a. Team-evaluation. This will be done by the team consensus.
   b. Student self-evaluation. Each student should honestly evaluate its contribution and project effort.

Bibliography

[1] **Ubiquitous Computing Fundamentals**
J. Krumm (ed.), CRC Press, 2010

[2] **Smart Sensors to Network the World**
D. E. Culler, H. Mulder, Scientific American, Jun 2004.

[3] **Designing the Internet of Things**
A. McEwen, H. Cassimaly, Wiley, 2014

[4] https://www.scrum.org/ (v. 28/09/2020)

[5] https://www.freertos.org/ (v. 08/09/2021)

[6] https://nodemcu.readthedocs.io/en/release/ (v. 28/09/2020)

[7] https://www.espressif.com/en/products/socs/esp32 (v. 23/09/2022)

[8] developer.arm.com (v. 28/09/2020)