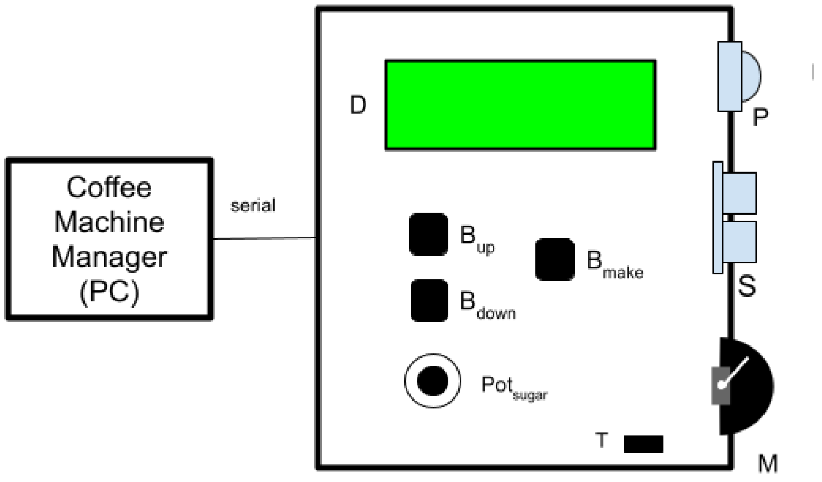
**Embedded Systems and IoT - a.y. 2021-2022**

# Assignment #2 - *Smart Coffee Machine*

We want to realise an embedded system simulating a smart coffee machine.

*Overview*

The system is composed of three tactile buttons $B_{up}$, $B_{down}$, $B_{make}$, a potentiometer $Pot_{sugar}$, a display D connected to the board through $I^2C$, a pir P, a sonar S, a servo motor M and analog temperature sensor T. The system is connected to a PC through a serial line. On the PC there is a simple application called "Coffee Machine Manager" that interacts with the system.



The coffee machine is able to make three different kinds of products: coffee, tea, chocolate. When refilled, the machine can make up to $N_{max}$ products for each type.

A user interacts with the machine by means of the buttons $B_{up}$, $B_{down}$, and $B_{make}$ – to select and make a product – and to $Pot_{sugar}$ to choose the sugar level. The making process is simulated by means of the motor M.

The machine has a power-saving modality, so that it goes to sleep if it is idle for a certain amount of time and no users are nearby.

The Coffee Machine Manager application is used to monitor and check the state of the machine, as well as to refill it when there are no more products that can be made.

*Detailed behaviour of the system*

- When the machine boots, it displays a welcome message (choose it) for a couple of seconds,  and initialises the number of products that can be delivered – $N_{max}$ items for each type.  After this initialisation stage, a message "Ready." is displayed on the display D.

- A user can (1) select the product to make by using  $B_{up}$ and $B_{down}$ and (2) start making process by pressing  $B_{make}$. The sugar level can be tuned by using the $Pot_{sugar}$.  When pressing $B_{up}$ and $B_{down}$ a message reporting the name of  currently selected product type is displayed on D. The message is kept for 5 seconds (if no other buttons are pressed), then the message "Ready." is displayed back.

- If/when the making process starts, a proper message "making a XXX…" is displayed (XXX = "coffee", "tea", "chocolate").  The making process of a product  is simulated by means of the motor M, rotating from 0 (beginning of the process) to 180 (end of the process, product ready). The making process takes $T_{making}$ seconds to complete. When the process has completed, a message "The XXX is ready." is displayed.

- A product is available for selection only if the current number of available items is greater than zero. If no product is available, then the machine enters in a "assistance" modality – displaying a message "Assistance Required" on the display D. Then, the normal behaviour of the machine can be resumed only by  means of the Coffee Machine Manager application – doing a refill in this case.

- When the product is ready, the product must be removed by the user. The product is considered removed as soon as the distance of the user as measured by the sonar S is greater than 40 cm. A distance less than 40 cm means that the product is still there and the machine has to wait for the user to take it. Nevertheless, if the user does not take the product within $T_{timeout}$ seconds, the machine goes on.  As soon as the product is removed or the timeout occurs, the motor M is reset to 0 (as fast as possible).

- When the machine is an idle state for more than $T_{idle}$ seconds and no user is detected (by the pir P) nearby, then the machine goes to sleep, waiting for some user to be detected

- Periodically, every $T_{check}$ seconds, the machine does an self-test – doing a complete rotation of the motor M from 0 to 180 and back (simulating some mechanic check) and checking the value of the  temperature Temp measure by means of the sensor T. If the Temp value is not in the range [$Temp_{min}$, $Temp_{max}$], then the machine enters in the "assistance" modality - displaying a message "Assistance Required" on the display D. In the "assistance" modality, no products can be made. The normal behaviour of the machine can be resumed only by  the Coffee Machine Manager application.

- The Coffee Machine Manager applications must provide a simple GUI (or event a textual UI, if preferred) that makes it possible to:
  - show the up-to-date state of the coffee machine:
    - modality, if it is idle or working, or assistance
    - the number of product items that are still available
    - the number of self-tests performed since the boot
  - refill - by means of a "Refill" button
  - recover - by means of a "Recover" button (useful in the case in which the machine has detected a problem during the self-test)

---

- Develop the embedded software on Arduino + PC connected through the serial line, implementing the Arduino part in C++/Wiring e the PC part in Java or in another favourite language. The Arduino program must be designed and implemented using task-based architectures and synchronous Finite State Machines.

- Some concrete values suggested for tests:

  $T_{making}$ = 10 seconds
  $T_{timeout}$ = 5 seconds
  $T_{idle}$ = 60 seconds
  $T_{check}$ = 180 seconds

  $Temp_{min}$ = 17 °C
  $Temp_{max}$ = 24 °C

For any aspect not specified, you are free to choose the approach you consider more useful or valuable.

**The Deliverable**

The deliverable consists in a zipped folder **assignment-02.zip** including two subfolders:

- **src** folder, including
  - two further folders: arduino and java, the former must contain a smart_cm folder, containing the Arduino project, and the latter the sources of the Java program
- **doc** folder, including:
  - a brief report describing the solution, in particular the diagram of the FSMs and the representation of the schema/breadboard – using tools such as TinkerCad or Fritzing or Eagle
  - a short video (or the link to a video on the cloud) demonstrating the system