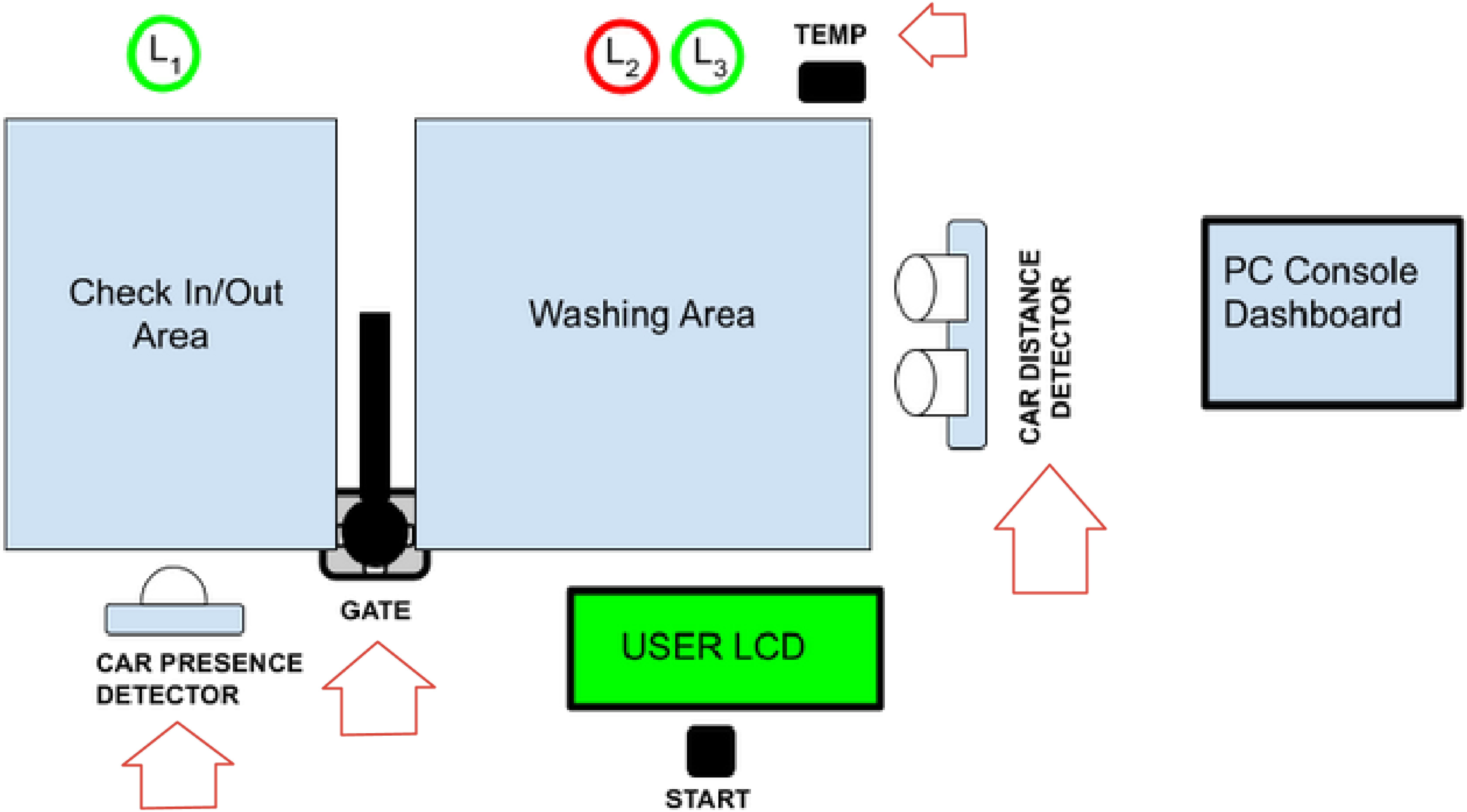

L1
L2
L3
TEMP
Check In/Out Area
Washing Area
CAR DISTANCE DETECTOR
PC Console Dashboard
GATE
USER LCD
CAR PRESENCE DETECTOR
START

Hardware components:

- CAR PRESENCE DETECTOR is a PIR, to detect the presence of vehicles in Check In/Out Area
- CAR DISTANCE DETECTOR is a sonar, to measure the distance of vehicles entering in the Washing Area
- GATE is a servo-motor, simulating a gate, between the Check In/Out Area and the Washing Area
- USER LCD is a I2C LCD, used to interact with users
- START is a tactile button
- L1 and L3 are two green leds, L2 is a red led
- TEMP is an analog temperature sensor

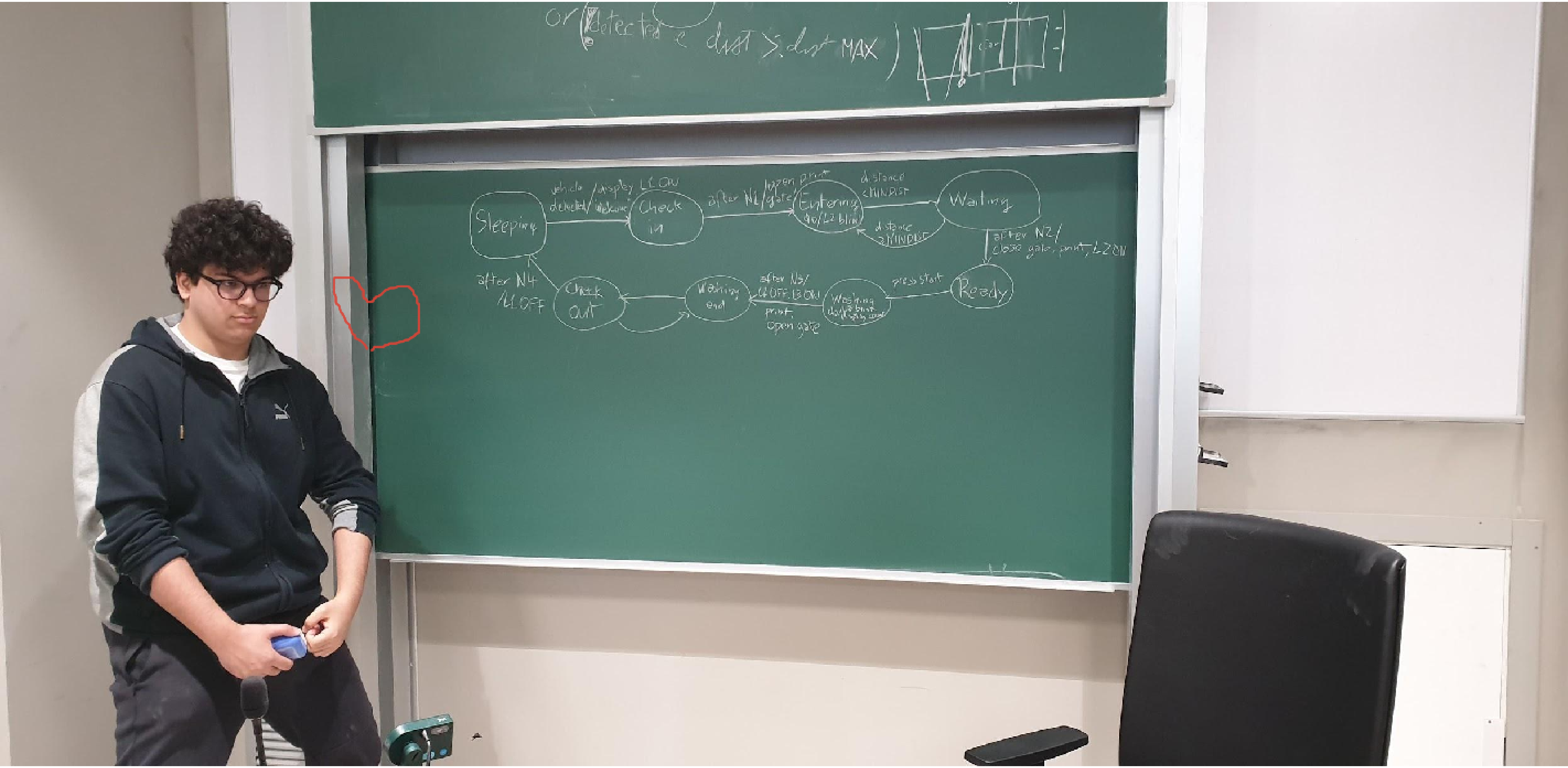
The embedded subsystem is meant to communicate by means of the serial line with a PC, running a PC Console Dashboard (for maintainers).

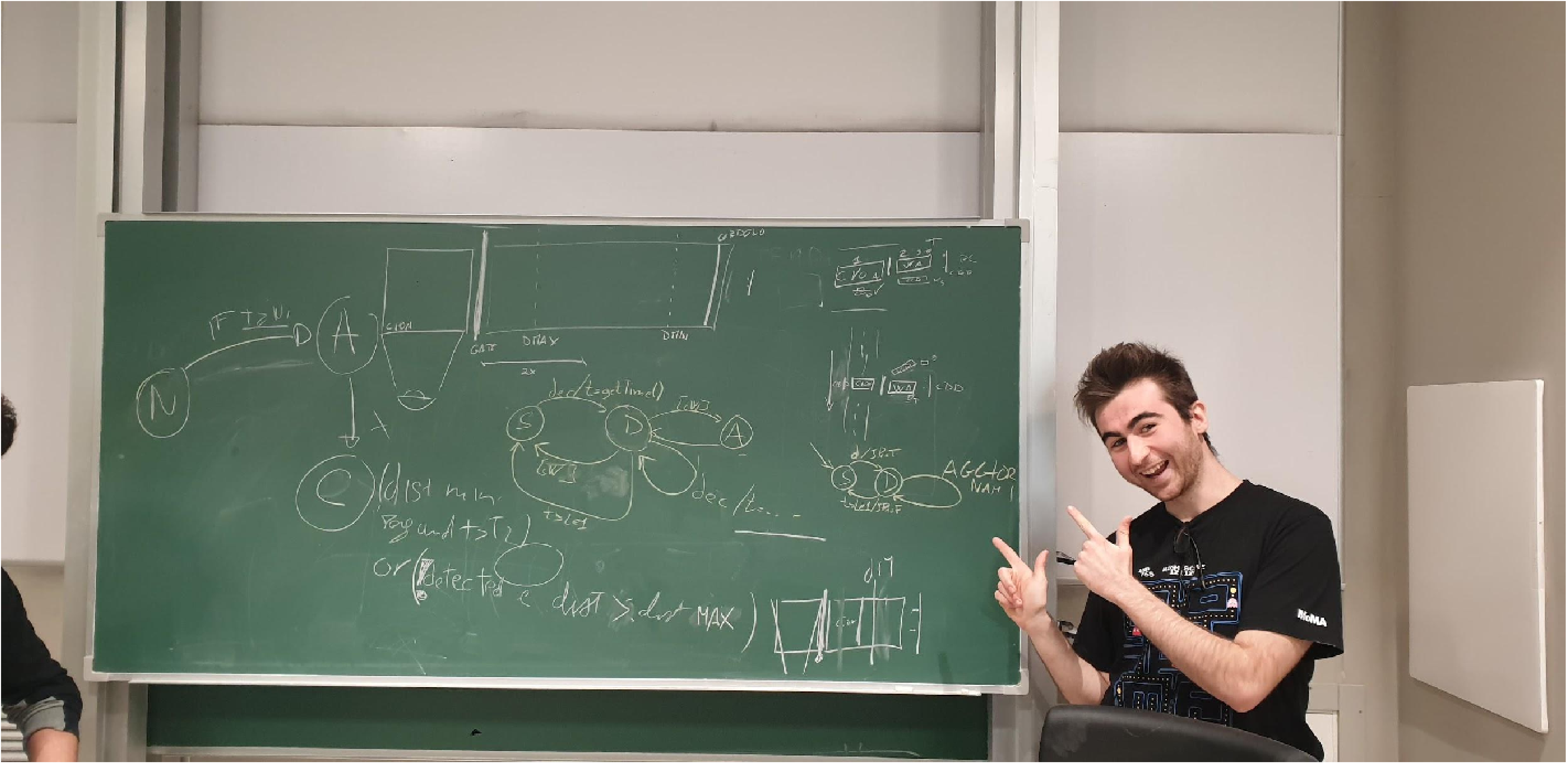
Description

- The simulated environment involves two main areas:
  - a Check-In/Out Area, welcoming vehicles that aim at being washed (one by one), and
  - the Washing Area, which is where the washing process occurs
- When no one is around, the system is off, sleeping.
- When a vehicle is detected by the CAR PRESENCE DETECTOR in the Check In/Out Area, the light L_1 is turned on and on the USER LCD the message: `Welcome` is displayed.
- After N_1 seconds that a vehicle is in the Check In/Out Area, the gate GATE is opened to allow the vehicle to proceed to the Washing Area and the light L_2 starts blinking with a period equals to 0.1 secs. On the USER LCD the message `Proceed to the Washing Area` is displayed.

- The entrance of a vehicle in the Washing Area is tracked by the CAR DISTANCE DETECTOR: when the measured distance is less than MINDIST for N_2 secs, the vehicle is considered fully entered. The gate GATE is then closed and the light L_2 stops blinking and is turned on. On the USER LCD the message: `Ready to Wash` is displayed.
- When the user presses the button START, the simulated washing process begins and lasts for N_3 secs. During this time, the light L_2 blinks with a period equal to 0.5 secs. On the USER LCD a countdown is displayed–either using numbers or a bar changing dimensions.
- After N_3 secs, the washing process ends: L_2 is turned off, L_3 is turned on, the message `Washing complete, you can leave the area` is displayed on the USER LCD and the gate GATE is opened to allow the vehicle to move away from the Washing Area. In this stage, the leaving process can be considered completed when the CAR DISTANCE DETECTOR measures a distance greater than MAXDIST for N_4 secs. At that point, the gate GATE is closed, L_3 is turned off

- The PC Console Dashboard is meant to be a simple GUI used by car-washing maintainers to monitor the state and functioning of the washing system and intervene if necessary.
    - In particular, It visualises the total number of washes done so far and the current state of the washing machine. To this purpose, during the washing process the temperature is monitored by means of the TEMP sensor, reporting the ongoing value on the PC Console Dashboard.
    - If the temperature becomes greater than MAXTEMP for N_4 seconds, then the washing process is suspended and a message `Maintenance required` is displayed on the PC Console Dashboard, along with the message `Detected a Problem - Please Wait` on the USER LCD.
    - Then, the PC Console Dashboard must provide a button `Maintenance done` to notify that the problem has been verified (and solved) and the washing process can go on.


or (detected & dist >. dist MAX)
Sleeping
vehicle detected/
/display L1 ON 'welcome'
Check in
after N1/gate
open print
Entering
do/L2 blink
distance <MINDIST
distance ≥MINDIST
Waiting
after N2/
close gate, print, L2 ON
Ready
press start
Washing
after N3/
L2 OFF, L3 ON
print,
open gate
Washing end
Check Out
after N4
/L1 OFF


IF t≥W1
N
A
C
GION
GATE
DMAX
DMIN
2x
dec/targetTime()
S
D
A
[dist min
Pay and t>T2]
or (detected c dist > dist MAX)
dec/t...
t>Let
d/SB.T
S
D
t>Let/SB.F
AGGTOR
NAH !
dir
CDD
MAX

Electron Serial port

Web Serial API

The Web Serial API can be used to access serial devices that are connected via serial port, USB, or Bluetooth. In order to use this API in Electron, developers will need to handle the select-serial-port event on the Session associated with the serial port request.

There are several additional APIs for working with the Web Serial API:

The serial-port-added and serial-port-removed events on the Session can be used to handle devices being plugged in or unplugged when handling the select-serial-port event. Note: These events only fire until the callback from select-serial-port is called. They are not intended to be used as a generic serial port listener.

ses.setDevicePermissionHandler(handler) can be used to provide default permissioning to devices without first calling for permission to devices via navigator.serial.requestPort. Additionally, the default behavior of Electron is to store granted device permission through the lifetime of the corresponding WebContents. If longer term storage is needed, a developer can store granted device permissions (eg when handling the select-serial-port event) and then read from that storage with setDevicePermissionHandler.

ses.setPermissionCheckHandler(handler) can be used to disable serial access for specific origins.

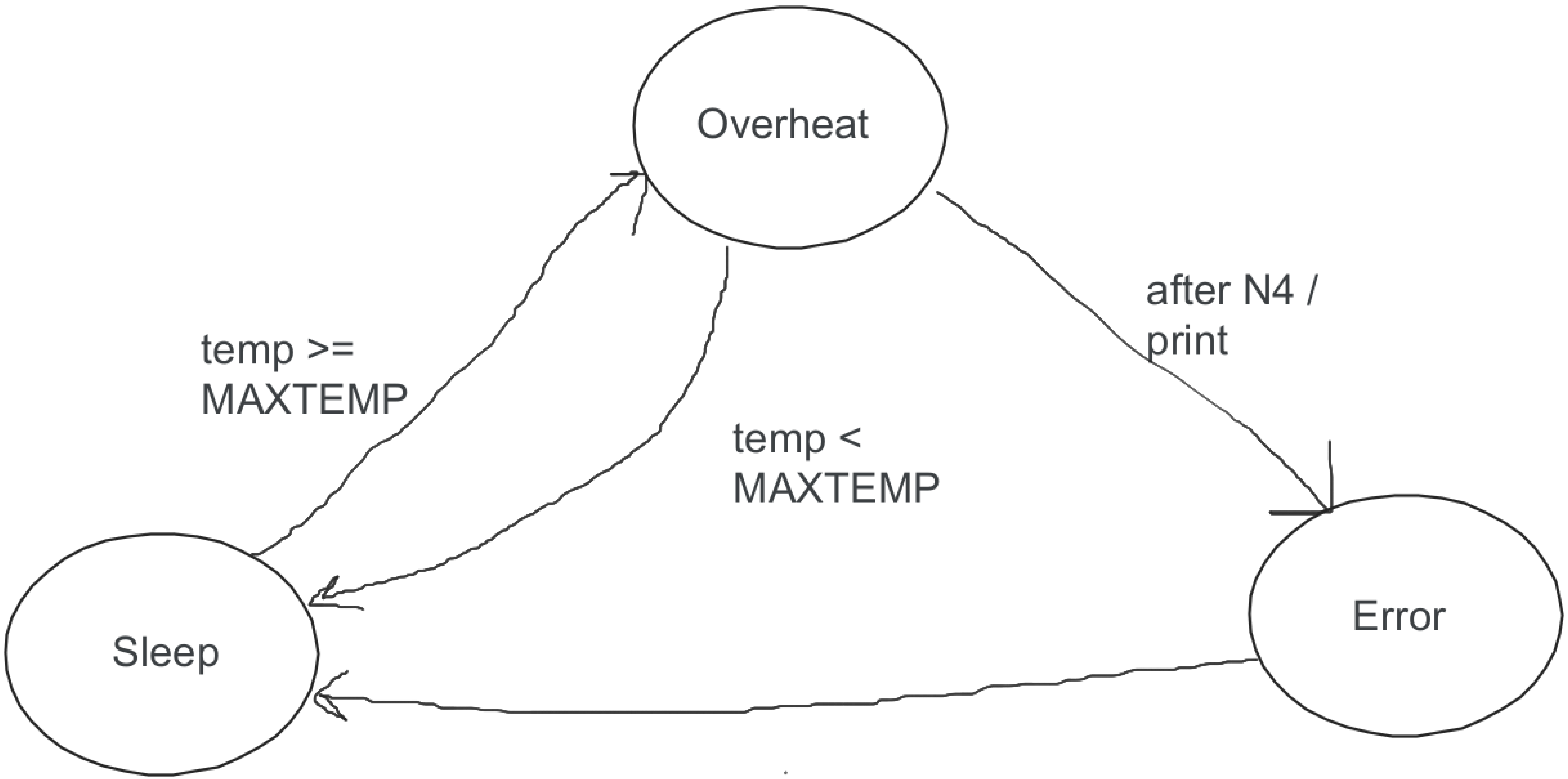
Example

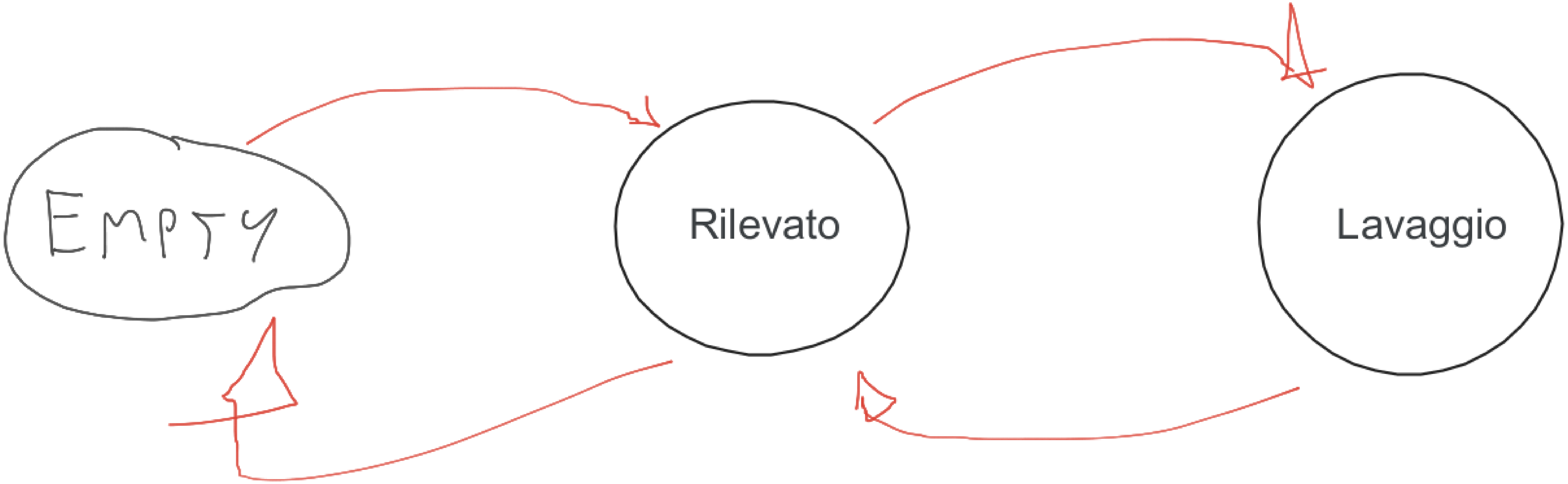
This example demonstrates an Electron application that automatically selects serial devices through ses.setDevicePermissionHandler(handler) as well as demonstrating selecting the first available Arduino Uno serial device (if connected) through select-serial-port event on the Session when the Test Web Serial button is clicked.

https://www.electronjs.org/docs/latest/tutorial/devices

jschlatt

Peack


Overheat
temp >= MAXTEMP
temp < MAXTEMP
after N4 / print
Sleep
Error


Empty
Rilevato
Lavaggio