1. Home automation, why ?

My first motivation, in chronological order, was to not let the terrace plants die during my holidays: the initial solution for a few years was to use irrigation timers bought at the consortium shop (50 €), but they often broke. Then I used a self-made IOT smart timer (€ 15), which can be controlled locally with a smartphone. Now, after the call from the concierge who told me about a pipe break while I was 100 km from home, the need to have remote control was also highlighted.

The second reason was the control of the room temperature. I have a centralized system, with radiators equipped with heat meters, and we don't like the cold. Therefore, the optimization of the management of the heating system aims both to improve comfort, to reduce costs and, more generally, to achieve a greener management, without unnecessary waste.

Third reason is security. I'm not particularly anxious about it, but a little bit of control over fires, gas leaks and intrusions I think is valid insurance.

A point often mentioned but which does not constitute a valid motivation for me (at least now) is the control of energy consumption and the use of "smart" appliances. They are probably very useful functions in the industrial or hotel sector, but they seem superfluous to me in my home, except perhaps to turn off the water heater when the iron is turned on, so as not to trip the automatic switch. But I have centralized hot water! (smile).

Concluding all these aspects, they can be faced and solved with a home automation control, better if integrated.

2. What is meant by integrated ?

For example, when you leave the house you generally set the burglar alarm on: at the same time it would be useful to lower the temperature of the radiators. The radiators must not be closed completely, because when you return they would take too long to restore the optimal temperature, but it is useless for them to go to their maximum when there is no one at home.
Conversely, in the case of a holiday weekend, it makes sense to completely turn off the radiators, to turn them back on a few hours before your return.

When you have two systems available (for example, 'alarm' and 'boiler thermostat'), albeit wonderful, from award-winning brands, very expensive but totally independent, you need to act on both separately each time.
Conversely, with a home automation system, which "integrates" alarm management and heating management, the two systems can easily exchange information, so a single action by the user can control both the alarm and the radiators, with greater simplicity and reliability.

These situations, or scenarios, are technically called 'use cases', and in current home automation terminology ‘scenes’. They represent an event and define the consequential actions to be taken.
The above can be represented with 4 scenes:

1) Leaving home for a short time
2) Departure for vacation (more than one day)
3) Imminence of returning from a vacation
4) Return home

A home automation system, therefore, must give the user the ability to define *their own scenes according to their specific needs*. This obviously can be done in various ways, and precisely the balance between the flexibility offered and the complexity of ***defining the scenes*** is an important evaluation factor of a home automation system..

For example, the definition of the scenes in IFTTT is simpler (and therefore more limited) than that offered by Tuya, which in turn has the limit of not allowing numerical calculations.
So not even with Tuya you can have a scene like this: “if the room is 12 ° C colder than the desired temperature, then turn on the air conditioner to heat it too”.

It is precisely these limitations (omnipresent for commercial reasons: the app or system must be as simple as possible) that create the need for extensions to the home automation system used, using methods that allow, where required, a freer definition of the scenes.

In some cases, moreover, it may be necessary to use devices not provided for in the chosen home automation system: this is another aspect of the extensibility of a home automation system.
For example, I might want to insert my actual timer to water the terrace in the home automation system: so I could condition the watering with weather information, obtaining savings in the use of water, while there would remain the possibility of autonomous operation in case of failure in the main system.
Another example: I have a small PM10 meter module, useful for assessing air pollution. It is a device with a USB serial output and I would like to insert it in the home automation system, as an alarm for the need for more ventilation.

Another important aspect of extensibility is the possibility of accessing all the data to have the possibility to process or record those of interest automatically and in the preferred format, whether they are ambient temperature and humidity, electricity consumption, etc. This function is generally not foreseen in commercial products, but it can be useful in many cases, for example to evaluate different solutions, for the comparison between successive periods, for the certification of maintenance within the limits of a given environment (I am thinking of greenhouses), for cost evaluation etc..

In conclusion, the ***extension of each home automation*** system covers various aspects, depending on the structure of the home automation system taken into consideration and on *the basis of the needs to be met*. This factor constitutes another important point of evaluation when we make choices.

3. Choice of a home automation system.

There are an infinite number of systems for domestic homes on the market, they are trendy. A first distinction can be made between two large groups:

- With devices connected to a control unit via a cable (‘wired’, with a BUS).
- With devices connected to the control unit via radio (various protocols: Bluetooth, Zigbee, WiFi etc.).

The first group includes generally proprietary solutions that cannot be integrated easily, developed by manufacturers of electrical appliances for domestic use, for example: BTicino, Gewiss etc.
The advantage is reliable and highly secure communication.
The disadvantages derive from the need to wire the BUS to all devices and from the fact that not everything can be wired (think of the alarm switches on doors and windows).
In addition, a limited choice of devices (typically mono-brand) and high costs are available.

The second group is in continuous development and consists of proprietary solutions, Open Source solutions, totally custom solutions, with a myriad of devices from cheap to very cheap.
The great advantage is the elimination of the need for *ad hoc* wiring, with a strong reduction in costs and installation complexity, especially in European homes almost all in brick and concrete.

The disadvantages are related to the lower reliability of radio connections, the risk of interference from malicious strangers, and the exposure of users to radio waves.

However, my idea is clear: in a new building, perhaps I would consider the idea of adding the tracks for a wired home automation system, but in an already built building I have no doubts, wireless solutions are preferred.

4. Why Tuya ?

In the myriad of wireless home automation solutions present, I chose **Tuya** for myself.

Tuya is a WiFi solution that occupies a particular place in the current commercial landscape. ***Tuya does NOT produce its own devices, but offers 2 services, one dedicated to hardware manufacturers, the other to users.***

Tuya offers ***manufacturers*** ready-made firmware, built using predefined modules (similar to how ESPHome does for ESP8266 / ESP32 based DIY projects). This solution cuts costs and time for new products. Tuya states that 5000+ devices use their platform (see https://expo.tuya.com).
The Tuya firmware offers an infinite number of options: connection using various channels: WiFi, ZigBee and others, OTA (on the air) update, user interface design, integration with Alexa and Google Home, etc... All continuously updated.

It offers ***users*** free use of a very efficient cloud, with 5 centers: China, East US, West US, Europe, and India:

- The response time in metropolitan areas in China is less than 40 ms (0.04 s).
- The response time in metropolitan areas in Asia is less than 80 ms (0.08 s).
- The response time in Europe and North America is less than 90 ms (0.09 s).

Naturally Tuya have created free apps (Tuya smart and SmartLife, *the latter preferred by me because it is enabled for Google Home****) which integrate all Tuya compatible devices into a single control application.***

A *win-win* solution, which is deservedly very successful: even established products, such as the simple *Sonoff-basic WiFi switch*, now have Tuya versions: no single manufacturer can compete..

Security in these systems is essential, both for end users and, above all, for manufacturing companies, which exert their influence in this sector to defend their investments: a good chance for users, who normally have little voice in this sector with big players (see whitepaper).

Some critical studies on the reliability of the solutions used can be found online, generally referring to obsolete versions of the Tuya protocols, now at version 3.3. In fact, the ability to OTA update the firmware of the devices with more recent versions is a guarantee for both the manufacturer and the end user.

A highly subjective question of principle remains open: my data externally? Do they contribute to BIG DATA? Sold on the dark web?
In my opinion, in this specific case, it is a marginal problem: it is neither personal data nor confidential information. Nothing comparable to ‘e-mails’. Usage statistics? Sure. Email address? I use one created only for Tuya, and besides, mine is public.
But above all I trust in the bargaining power of the various producers, Tuya's real customers, who are also in competition with each other.

5. Integrations and customizations

*All right then with **Tuya**? Yes of course, but also no.*

All good if our needs are met with the performance that Tuya devices and apps offer. These are the highest performances currently available on the market, inimitable in terms of ease of installation and use, economic both in terms of cost and activation time, with a very complete range of devices, produced by many companies also in competition with each other, all upgradeable: an ecosystem with a long time horizon ahead.
The individual devices can be easily integrated with each other with the automation tools provided by the apps: scenes and automations. It is trivial with Tuya to connect the temperature control with the alarm system and create scenes such as 'leaving home' and 'beginning of vacation'. With the geolocation (available with *smartlife*) you can even have a fully automatic operation.
Why reinvent hot water?

*However, we may have more advanced but legitimate needs, which require additions to overcome the limits of a home automation system built using **devices + tuya cloud + app**.*

Some examples based on my personal needs:

1. I might want to make better use of the available data: for example, in the climate field, *smartLife* offers daily, monthly and yearly temperature graphs, for each probe or radiator, but not the possibility to export this data for further analysis, even just for calculate an average.
2. The automation mechanisms, defined as ‘scenes’ (also ‘Tap-to-Run’) or ‘automations’, depending on whether they are activated: by the user or by events from the various devices, are among the most powerful available in home automation applications. However they are simple if… then…, no calculation is possible. I would like to try more sophisticated algorithms, AI etc ..
3. A burglar alarm made with Tuya ceases to work if connections with the cloud are lost: it seems important to me to be able to increase the reliability of any alarm system so that they work even in the event of isolation.
4. Tuya cloud only controls Tuya devices, because only for these it receives royalties from manufacturers, but we may be interested in including information provided by other equipment, not (yet) present in the Tuya ecosystem, or already in ours, into our system possession, or simply, use information obtained from the internet (e.g. weather forecast: it can be a waste to water the plants if it rains tomorrow).
5. Other times I design and build new devices myself (DIY) either because they are new and not commercially available, or because I have a better solution in mind than the existing ones. Tuya offers the possibility of prototype development, but it is expensive and requires the use of their modules, at least for the communications part
6. Tuya apps (*smartLife and tuyasmart*) are nice and easy to use. But what if you wanted a different UI? For example, a graphic map of the environment with information on where the various devices are located? With status information and clickable icons? On smartphone or on PC?
7. With the cloud, Tuya devices can be controlled by apps from all over the world. Beautiful! But can I extend this feature to *my non-Tuya* devices too?
8. The extension to *Google home* and therefore having a *voice interface* is very simple with *smartLife*, so I can give voice commands to my home automation system. The predefined commands are quite elementary (often only ON/OFF), but all the scenes defined in *smartlife* can be activated vocally: the commands not present on Google are therefore achievable in

> this way (example: "Hi Google, run 'tune tv on Raitre'"). Even more beautiful! But can I extend this feature to *my non-Tuya* devices too?

These (and others: find yours...) are the reasons that can push a user to look for a methodology of integration between *Tuya* and the *outside world*. It is a problem made not easy to solve precisely because of the protocols with which Tuya protects the security of its devices and its cloud.

6. Why node-red ?

Integration, okay, but with which outside world?

For me the answer is really simple: I don't want another cloud (certainly worse than Tuya, eg IFTTT, not free from 2020/10/08) or another home automation application, even if Open Source, nor do I want to change the firmware of Tuya devices, so as not to use the Tuya-cloud .
At the dawn of home automation it might have made sense to build your own system from scratch, but now? The Tuya system works really well as it is, while the effort to change the firmware of the devices, for example to use *Tasmota* and *MQTT*, is certainly limited in the simplest cases, for example switches, but let's consider my radiator thermostats with 28 parameters (data points) and with Zigbee communications, or the firmware of a washing machine: the commitment can be truly remarkable. *Is worth*?

From my point of view, DIY is only worthwhile if it is something that is not achievable with the devices on the market, usually really convenient: my energy and my time have a cost. *Inventing the well? No thanks!*

> *All the problems that the use of **Tuya** can present can be solved, in my opinion, by **integrating Tuya with an open environment**, which simplifies the integration with third-party devices and solves the problems of the various communication protocols, which allows easy access to the information resources on the network and which enable the implementation of recovery strategies in the event of malfunction.*

Furthermore, this environment must offer a simple use of DB and HTML interfaces (it is the universal solution) and of course it must be programmable, to adapt to any scenario, but with minimal effort: *I have no time to waste*.

node-red is the ideal candidate, it offers rapid graphical programming, with an impressive set of ready-made blocks (custom-nodes) but at the same time remains programmable at low level, where necessary, using js. ***Node-red*** is now an Open source project, born in IBM, with a large and lively community, truly perfect for these custom applications, with a low learning time.
Many custom home automation systems are made with ***node-red***.

7. *SmartLife <=> node-red integration: tuyaDEAMON*

*Identified the problem, "**node-red<=>tuya communications**", I looked at the available solutions..*

I did not find a 'solution' but good tools (tuyapi and node-red-contrib-tuya-smart-device), so I designed and implemented a project, tuyaDAEMON (now in alpha test, see https://github.com/msillano/tuyaDAEMON) to isolate and solve all problems

tuyaDAEMON can be used in different ways:

1. configuration of a new Tuya device, compatible with *node-red-contrib-tuya-smart-device*:
   - The atomic information (called 'data points', identified by a number: 1, 102...) are identified and renamed by choosing user-friendly terms (e.g. 102 = 'temperature').
   - All the operations that ***smartLife*** allows to carry out with the device can be easy identified in this way.
   - The commands that the device accepts for each data point are then identified through trial and error.
   - You update the configuration file with this information, and forget it: *tuyaDEAMON* does all checks.

2. a particular HW and software combination allows you to define and use bidirectional TRIGGERS. The use of tuyaTRIGGER module completes *tuyaDAEMON* allowing:
   - to read and write the data points of those Tuya devices that cannot be reached directly with *node-red-contrib-tuya-smart-device* (limited to BOOL, ENUM, and events)
   - to perform *automations* in *tuya-cloud* commanded by *node-red*
   - to run *flows* on *node-red* from a *scene* activated by the user on the smartLife app
   - to run *flows* on node-red activated by *automations* via *tuya-cloud*.
   - to run *flows* in node-red with *voice commands*.

3. extension of the *Tuya home automation system*:
   - All data read by *Tuya* devices configured on *tuyaDAEMON* are available in an updated RT global structure, which uses only user-friendly names: the data can easily be read from other flows, for automations and user interfaces etc. (e.g: `global.alldevices.["Thermometer # 2"]. Temperature = 19`).
   - A selection of all communications is stored in a mySQL DB table, which can be consulted, even by non node-red applications (e,g, php), for control, for historical and statistical analysis etc.
   - *TuyaDAEMON* also manages "_system", a *'fake' device* which provides information about *tuyaDAEMON* itself: currently connected devices, alarms for lack of WiFi connection etc.
   - Other *'fake' devices* can be used to handle custom devices (not Tuya compatible) via *node-red*.