Università degli Studi di Salerno
Internet of things – **P. Ritrovato**
A.A.: 2019 - 2020

Pill Notify

Emanuele Triuzzi `06127 04000`
Lia Trapanese `06127 04183`

What’s about

The system allows to manage the need for medicine assumption by a person with problems such as dementia, amnesia and difficulties in the conception of time

In this way:

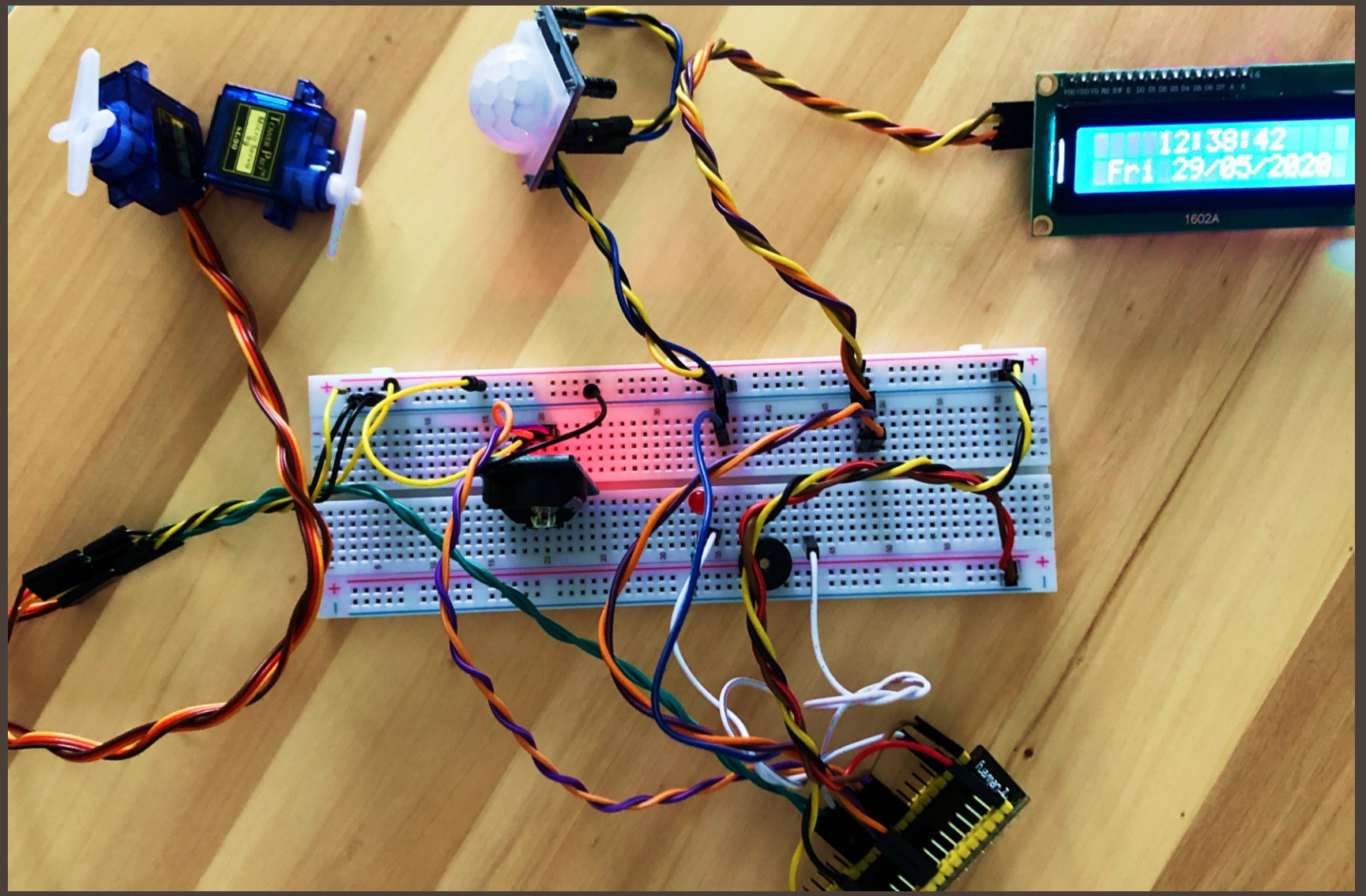
- You avoid forgetfulness and prevent mistakes in following a medical prescription that involves taking medicines at specific times on specific days
- Allow control over the taking of a medicine remotely by individuals close to the patient as family members or by the attending medic

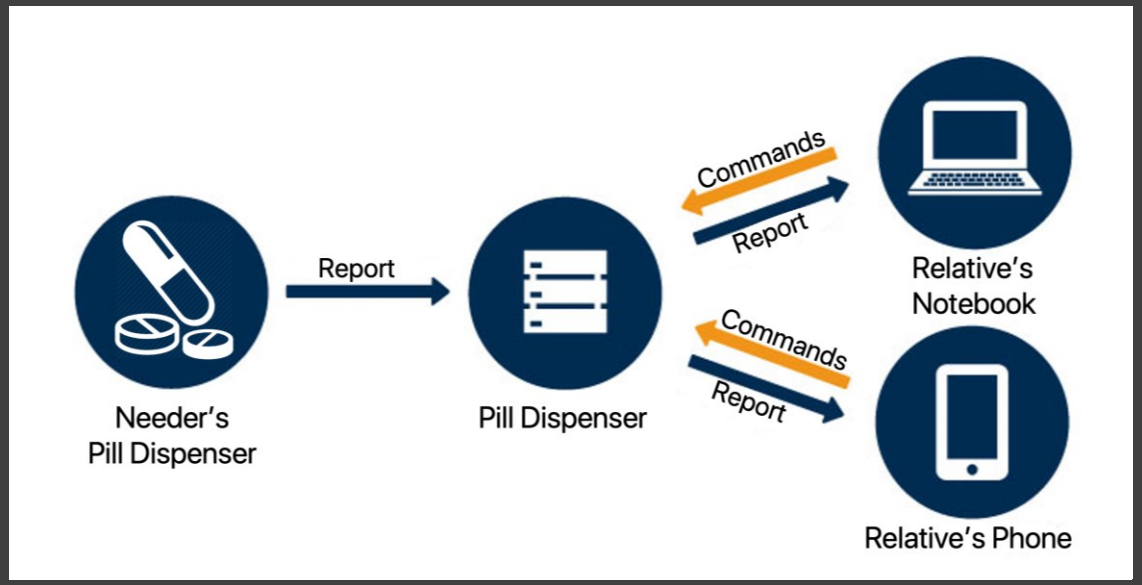
Implementation

In an ESP32 DevKitC flash memory is stored a time-table for each weekday, associated with the pill prescription

When it's assumption time for the current day servos drop a pill, and sets up an audible and visible alarm

The alarm is turned off by a movement of the needer's hand in the pill compartment by a PIR sensor





In our case, PillNotify works on two topics:

PillNotify/Needer: Relatives can add and edit the pill schedule, the system will adapt based on those comunications

PillNotify/Relative: The system publishes reports about the assumption of the pill, so relative can be notified in case of unexpected behaviour

MQTT Protocol

MQTT is a protocol that allows edge-of-network devices to publish to a broker on a specific topic

Clients connected to the broker and subscribed to topics will receive new messages published

The broker mediates communication between devices, when a client publishes on a topic, the broker forward te message to the clients that are subscribed

I²C Protocol

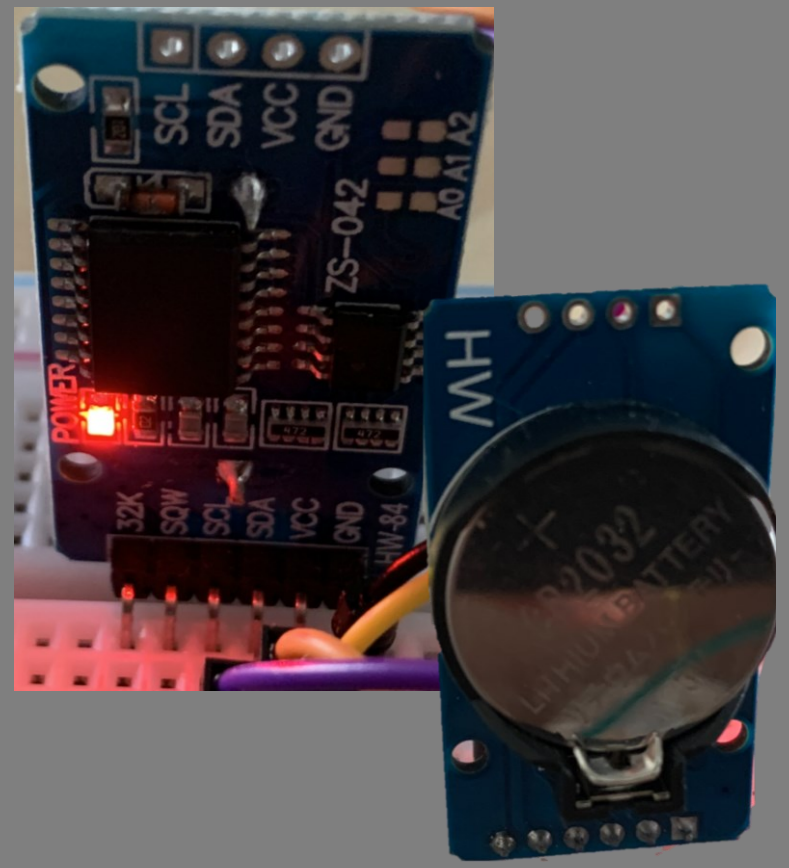
Pill Notify works with two I²C interfaced devices

I²C is a serial transmission protocol, so data is transferred bit by bit along a single wire *"DATA"*

I²C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master

Lcd Display

Real Time Clock



Flash Memory

The data saved in the flash memory remains there even when the ESP32 resets or when power is removed

The flash memory is very similar to the EEPROM. Both are non-volatile memories

One limitation of the flash memory is the number of times you can writes data on it: most devices can write up to 100’000 – 1’000’000 times.

This limits the life expectancy of the project to ~350 years which shouldn’t be a problem

JSON example stored on flash memory

```
{
    "name"      : "Maurizio",
    "pin"       : 123456,
    "pillName"  : "Pressione",
    "count"     : 5,
    "monday"    : [ [12, 0], [18, 0] ],
    "tuesday"   : [ [15, 0]          ],
    "wednesday": [ [ 8, 0], [18, 0] ],
    "thursday"  : [ [15, 0]          ],
    "friday"    : [ [17,30]          ],
    "saturday"  : [ [20, 0], [23, 0] ],
    "sunday"    : [ [12, 0], [18, 0] ]
}
```

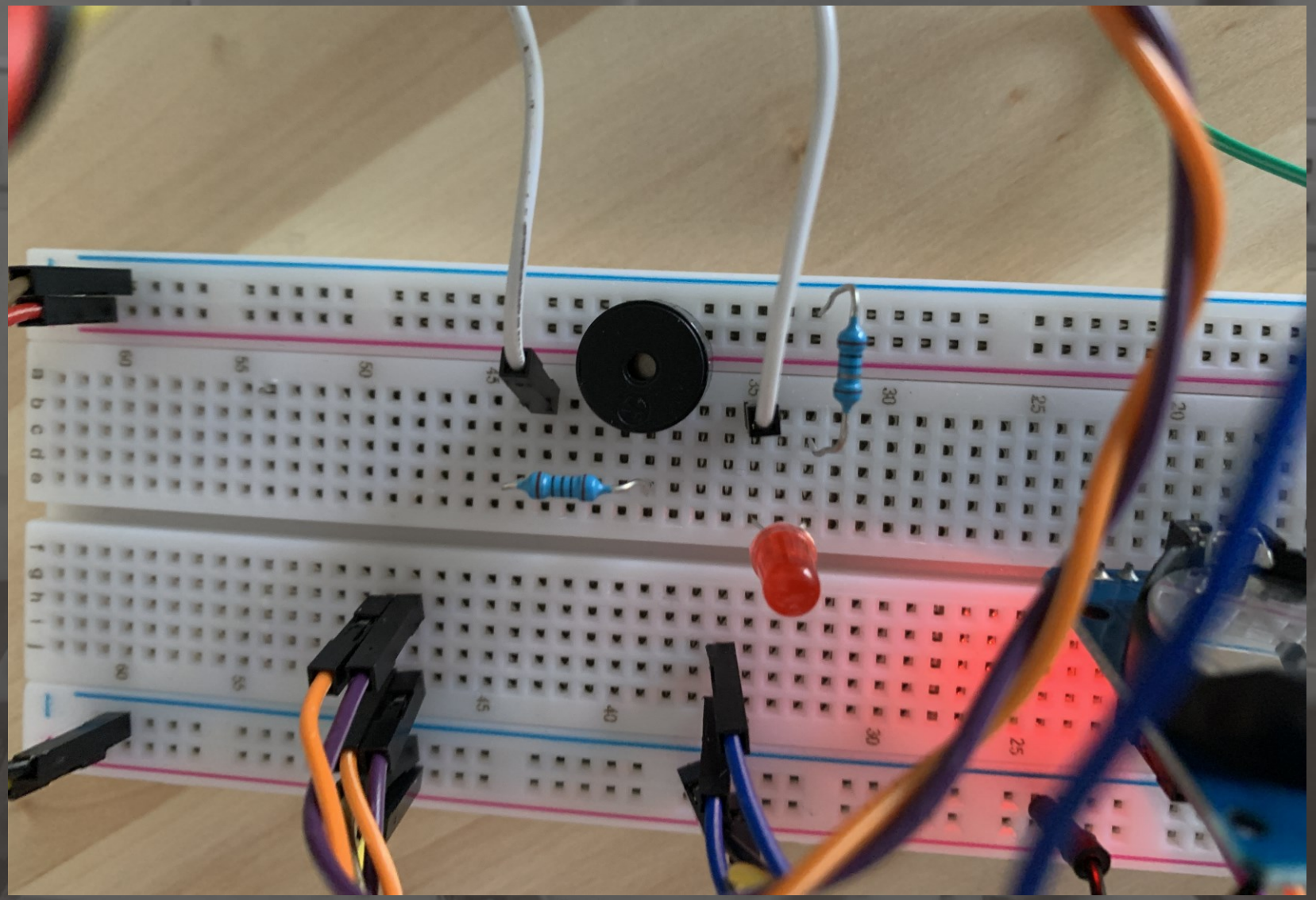
In our case, we store informations about the needer, useful informations to authenticate via MQTT, and to use and update the user pill’s time table

Alarm Event

When the time of the real clock corresponds to one of the times associated with current weekday, an alarm is triggered

The alarm consists in a tune played by the buzzer and a blinking led

The event can be turned off only when the needer take the pill, so when the PIR sensor perceive his hand in the apposite compartment



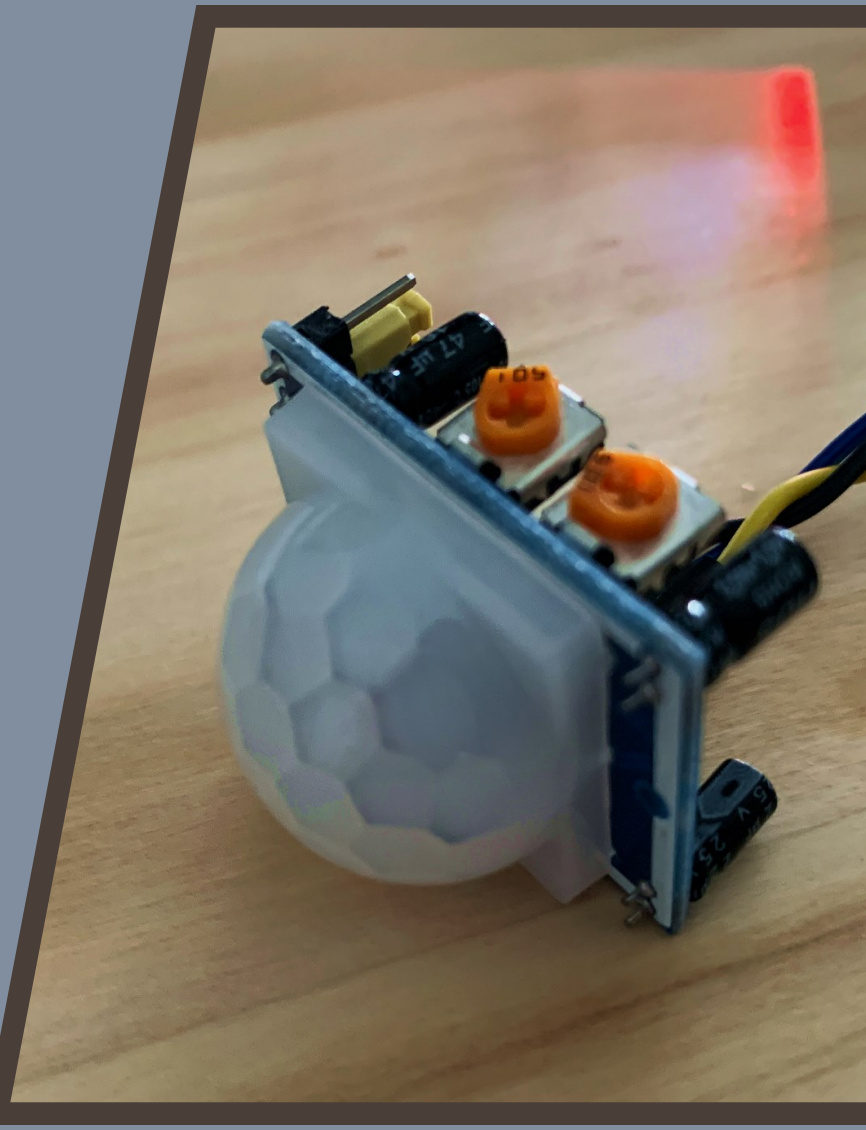
PIR Sensor

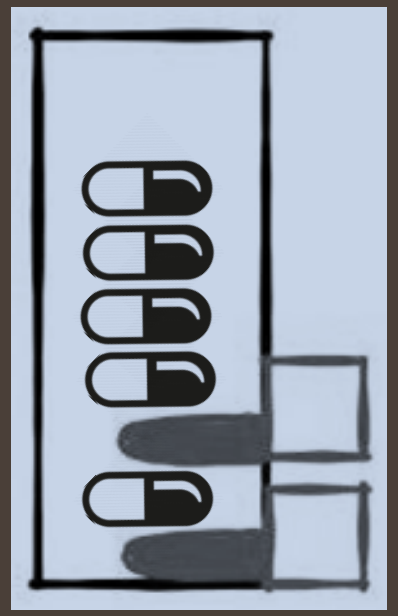
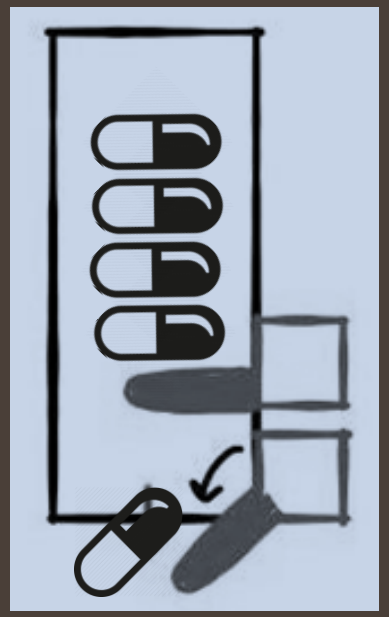
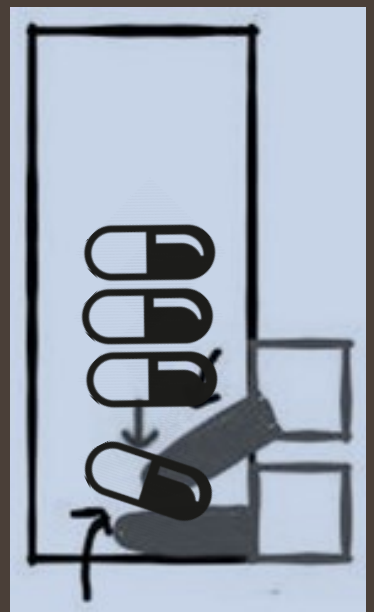
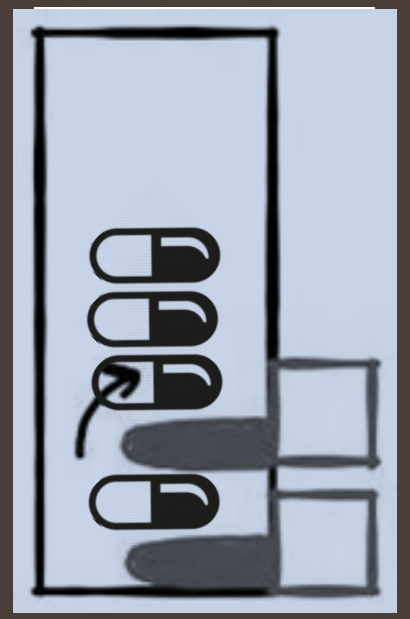
It is an electronic sensor that measures infrared (IR) rays radiated by objects in its field of view. For this reason are usually used as motion sensors.

All objects with temperatures above absolute zero emit energy in the form of light radiation. Most of the time these radiations are invisible to the human eye, since at a frequency lower than that of the light of the visible spectrum, but they can be detected through specific electronic devices designed for this purpose.

The term passive refers to the fact that PIRs do not emit energy in any form but work exclusively by detecting the energy released by objects.

In our case, the PIR is responsible for perceiving the presence of the hand in the appropriate compartment where the pill is supplied.



PWM

The Pulse-Width-Modulation allows to obtain a variable average voltage depending on the ratio between the duration of the positive and negative pulse (duty cycle)

With this technique, devices can be controlled with great precision. In this project, the buzzer is controlled by PWM to reproduce the alarm track, the same applies to the LED and the servo motors, creating the pill delivery mechanism

Multithreading

The system works through the harmony of different threads. In addition, there is an event, "alarm event", which is set if it is the time of administration of the pill and is used to activate the alarm

- **Alarm Thread (AT)**: Actually the Main Thread after the system initialization phase. Drops the pill when the alarm is set on, reads from the PIR sensor to set the alarm off if the pill is correctly taken

- **Time Thread (TT)**: Checks the timetable stored in the flash memory to check if the current time, read by the clock, corresponds to the times of the current weekday

- **LCD Thread (LT)**: Updates each to second to display the current time and date, if it's time to take the pill, it shows an indication message to the user

- **MQTT Thread (MT)**: Checks and analyze the message published on the PillNotify/Needer topic. If that's a JSON message, checks if it's addressed to you, performs an authentication and executes the function requested by the client. Moreover, sends report about pill assumption and the amount of remaining pill in stock

- **LED & Buzzer Threads (LBT)**: Let the LED blink and makes the buzzer sound when the alarm event is on.

MQTT Functions

An important feature of the system is the possibility of checking and following medical prescriptions: the user or a family member can send messages to modify the timetable in order to adapt the assumption of the pills, refill the number of available pills, and change the current pill.

The main functions are:

- **Set schedule**: Sets the pill assumption times (hours:minutes) for each specified day/days, multiple days can be specified via the “even”, “odd” and “all” words

```
{"name":"Maurizio","pin":1234,"pillName":”Pressione","function":"set_schedule","arg0":["all", [[10,0]]}
```

- **Remove schedule**: Remove the current stored schedule of the specified day/days

```
{"name":"Maurizio","pin":1234,"pillName":”Pressione","function":"delete_schedule","arg0":"wednesday"}
```

- **Refill stock**: Updates the number of pills available in the system. This number is used to communicate to the relatives if the stock is running out of pills

```
{"name":"Maurizio","pin":1234,"pillName":”Pressione","function":"refill_stock","arg0":6}
```

- **Change pill**: If the patient must change pill, this function updates the stored name in flash memory and sets the current pill count to zero

```
{"name":"Maurizio","pin":1234,"pillName":”Pressione","function":"change_pill","arg0":"Cuore"}
```

Other features

The system therefore, thanks to the authentication and the structure of the JSON saved in flash memory, is capable of supporting different users, multiple pills for the same user and guarantees access security to prevent abuse of the MQTT functions.

Moreover, the system can also work offline, obviously the functions are going to perform until the connection to the MQTT broker.