1 Introduction

In this project a real-time application is developed to recognize smells from an artificial nose. The sensor used for the application is an air quality gas sensor.

The rest of the documentation is structured in the following way: in section 2 the tasks are explained one at a time, in the section 3...

2 The tasks

In our application we have 5 periodic tasks (Figure 1): graphic task, sensor task, neural network task (made with Tensorflow), keyboard task and the store image task.

The main function sets everything up for the tasks, except for the store image task. The keyboard task is in charge to execute the store image task when the `ENTER` key is pressed. If the store image task is already in execution and the `ENTER` key is pressed this it's terminated.

Before start the store image task it's possible to write the name of the directory in which the images will be saved; if no name it's writed the images will be saved into *image_neural_network* directory.

The sensor is readed by an Arduino M0 pro; the sampled data readed by arduino are sent via the serial port to our application and readed by the sensor task. All the tasks are terminated by the keyboard task when the user presses the `ESC` key.

2.1 Main function

In the main function [1] all the tasks, except the store image task, are started and the mutexes initialized. The mutexes are two, one for the data readed from the sensor and the other for the results given by the neural network. The main also starts allegro and waits for the termination of the keyboard task. Once the keyboard task terminates the main cancels all other task and wait for their termination.

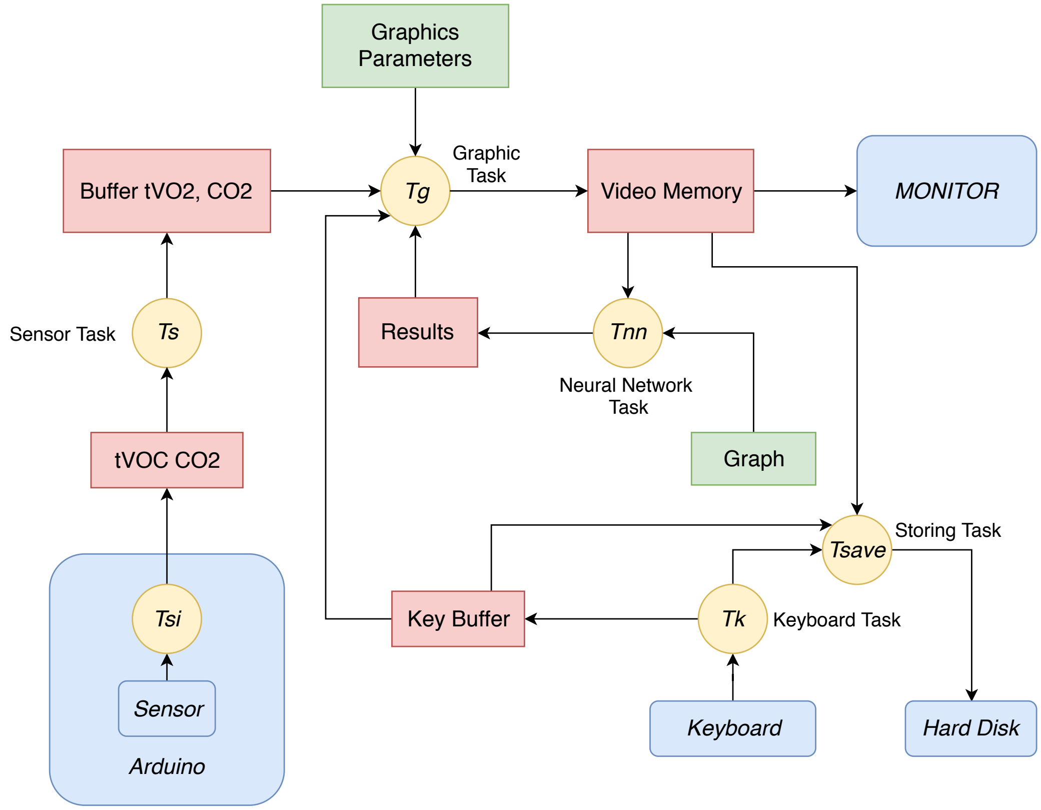

Figure 1: Task diagram

2.2 Graphic Task

The graphic task prints the interface of our application. The interface is divided into different areas that contains for each one the graph, the image, the results, the legend, the current values and the current status (`WRITING` or `SAVING`) followed by, if present, the keyboard input.

2.3 Sensor task

The sensor task [3] reads the values from the arduino which it sends on serial port the values given by the sensor. The readed values are stored into an array and used by the graphic task to draw the image and the graph; the current readed values are also standalone printed by the graphic task.

Algorithm 1 Main

```
$T \leftarrow$ *tasks to be started*
Mutexes and allegro initialization
for $t \in T$ do
    start $t$
end for
loop wait for termination of keyboard task
end loop
for $t \in T$ do
    cancel and join $t$
end for
```

Algorithm 2 Graphic task

```
$p \leftarrow$ *task period*
set activation task
draw interface background
loop
    draw graph
    draw image
    draw results given by neural network
    draw current values readed from sensor
    draw keyboard input
    wait for next activation
end loop
```

2.4 Neural network task

The neural network task [4] recognizes the smells using the current image created with the value sampled from sensor.

2.5 Keyboard task

The keyboard task [5] takes input from keyboard and put it into *keyboard buffer*. The keyboard buffer is printed by the graphic task in its area of interface. The string contained into keyboard buffer is the directory, under *image_neural_network*, where the images are saved by the store image task. The keyboard task shall be in two different mode: `WRITING` or `SAVING`. The

Algorithm 3 Sensor task

```
p ← task period
initialization data_q
initialization serial port
set activation task
loop
    data_q ← data_q + values readed from sensor
    wait for next activation
end loop
```

Algorithm 4 Neural network task

```
p ← task period
Tensorflow initialization
set activation task
loop
    image ← current image
    results ← use neural network with given image
    wait for next activation
end loop
```

task is started in `WRITING` mode. During this mode it's possible to write the name of the directory in which the images are saved. The name can contain letters, numbers, minus, underscore and point; it's also possible to delete the written characters pressing the `BACKSPACE` key. Pressing the `ENTER` key, the current mode is switched from the `WRITING` to the `SAVING` mode or vice versa. When the `ESC` key is pressed the task terminate causing the closing of our application.

2.6 Store image task

The store image task [6] is activated/terminated by keyboard task when the `ENTER` key is pressed. Once the task is activated the image are saved every 300 milliseconds. The images saved by this task are used to train the neural network for the recognizing of smells.

Algorithm 5 Keyboard task

```
p ← task period
cur_mode ← WRITING
create key_buffer
keyboard initialization
set activation task
repeat
    key_pressed ← key code from keyboard
    if key_pressed == ENTER then
        if cur_mode == WRITING then
            cur_mode ← SAVING
            start store image task
        else
            cur_mode ← WRITING
            stop store image task and clean key_buffer
        end if
    else if cur_mode == WRITING then
        if key_pressed equal to letter, number, minus or point and key_buffer not full then
            key_buffer ← key_buffer + key_pressed
        else if key_buffer == BACKSPACE and key_buffer not empty then
            remove last element from key_buffer
        end if
    end if
until key_pressed != ESC
```

Algorithm 6 Store image task

```
p ← task period
dir ← path to directory where images are saved
set activation task
loop
    save image to dir
    wait for next activation
end loop
```