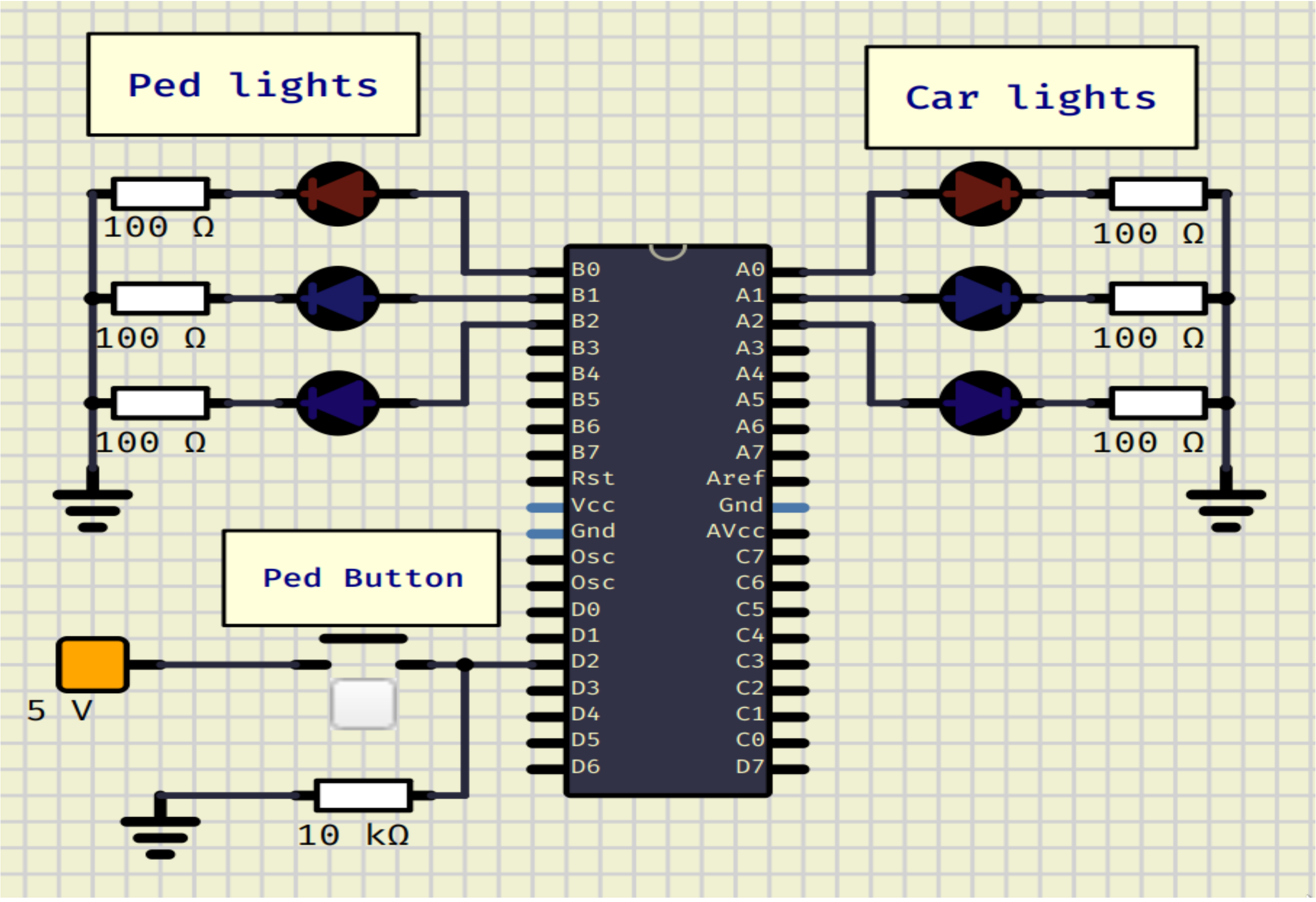
System discription:

It is an implementation of a traffic lights system with an on-demand crosswalk button.

The system consists of 3 lights (red, yellow, green) for both perdesterians and cars, and a button of on demand crossing for pedesterians.

Hardware:

- atmega32.
- 2x(red, green, yellow) LED.
- push button.
- 6x(100Ω) and 1x(10KΩ) resistor.



System constraints:

components :

- atmega32
- (rgb) led
- push button

system must have a 15s cycle switching between red → yellow → green → yellow → red …

push button activates pedestrian mode for 1 cycle.

system design

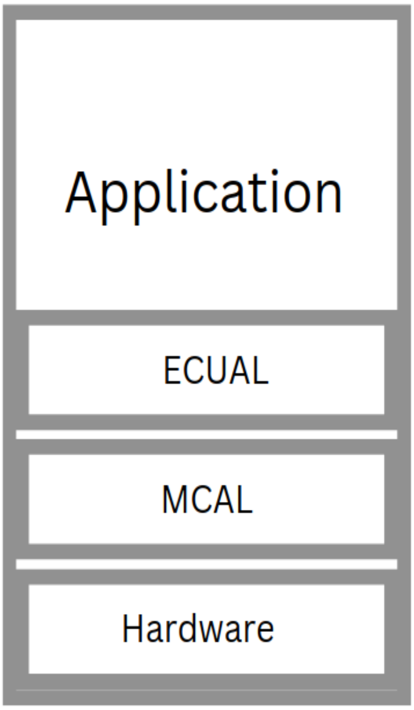
System is a minimal with layers to handel every stage and drivers for every controllable component.

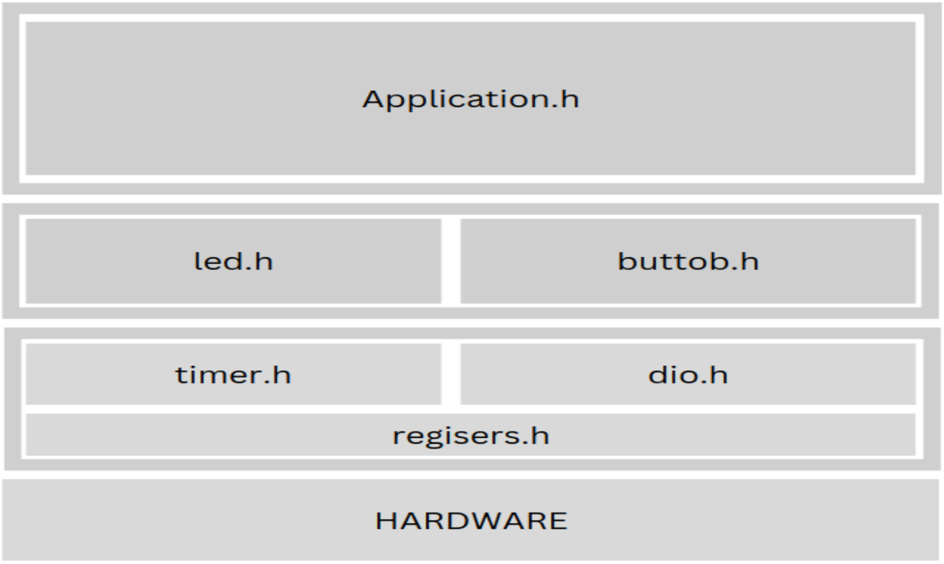
The software is seperated into 3 layers:

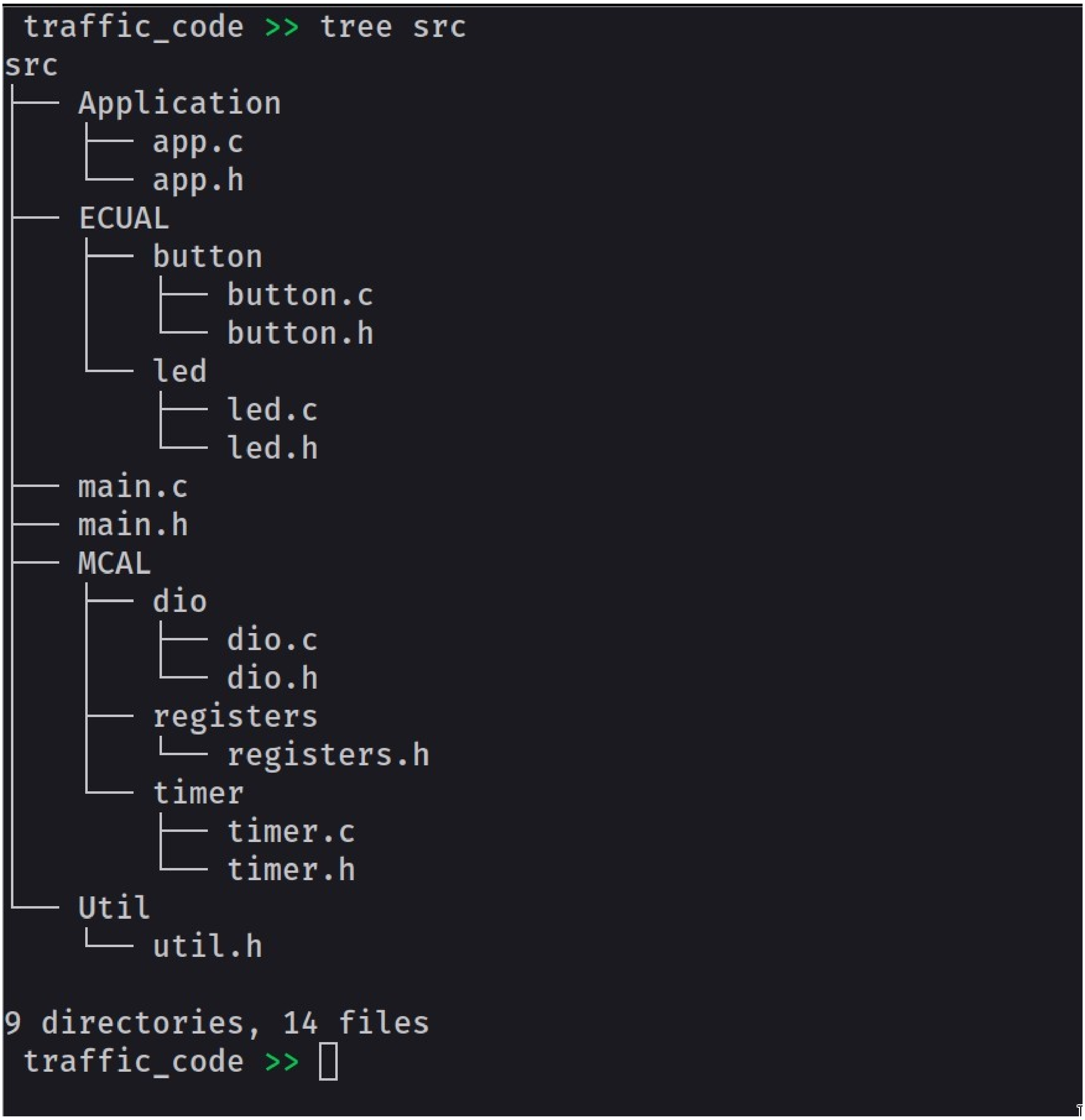
MCAL comunicates with the hardware directely with the registers.

ECUAL is the drivers layer, it only comunicates with the MCAL layer and is independent from hardware.

Application layer is the interface, it only comunicates with drivers and is depedent from the previos layers.



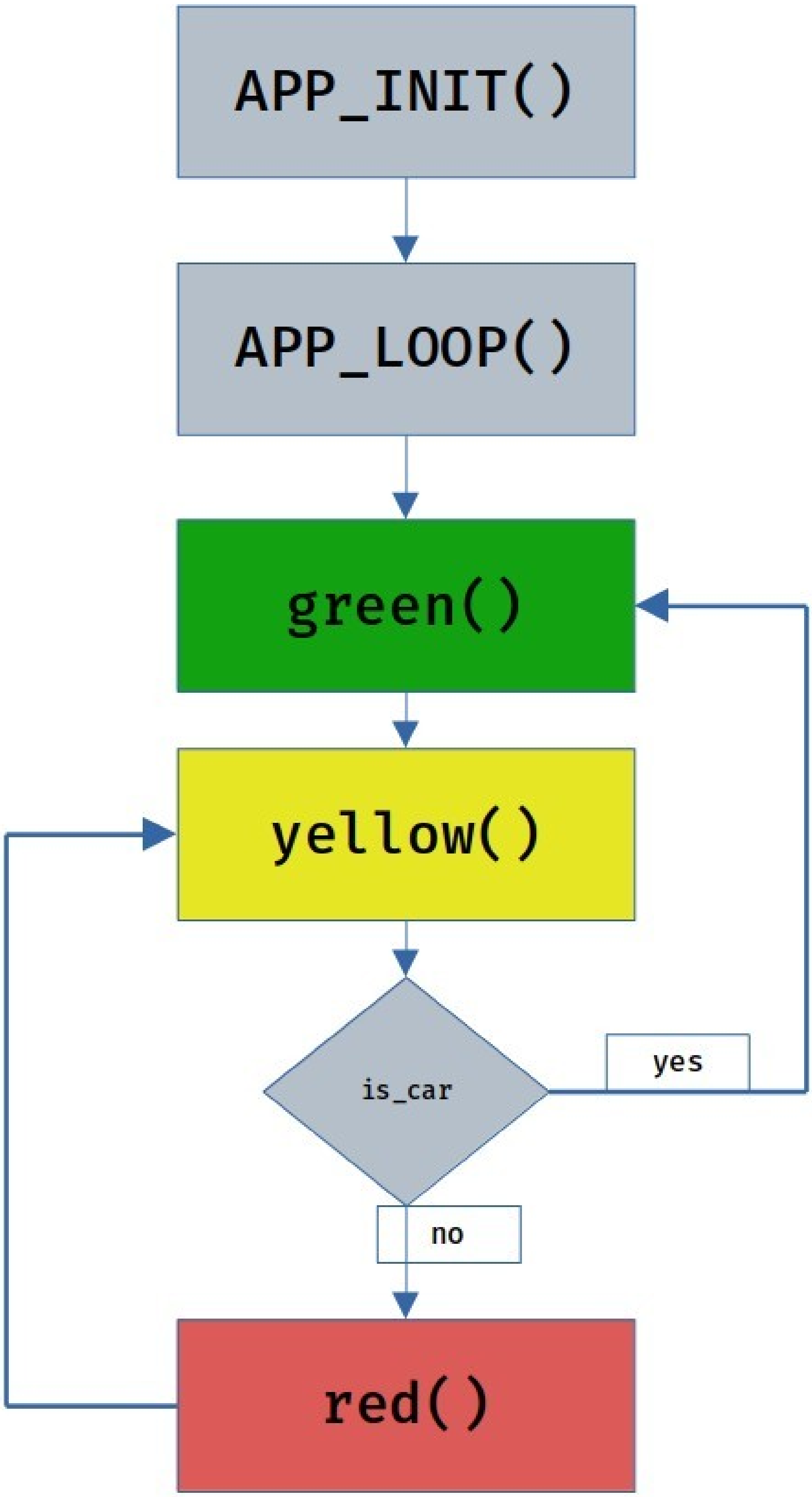




```
 traffic_code >> tree src
src
├── Application
│   ├── app.c
│   └── app.h
├── ECUAL
│   ├── button
│   │   ├── button.c
│   │   └── button.h
│   └── led
│       ├── led.c
│       └── led.h
├── main.c
├── main.h
├── MCAL
│   ├── dio
│   │   ├── dio.c
│   │   └── dio.h
│   ├── registers
│   │   └── registers.h
│   └── timer
│       ├── timer.c
│       └── timer.h
└── Util
    └── util.h

9 directories, 14 files
 traffic_code >> 
```

Simple flowchart


APP_INIT()
APP_LOOP()
green()
yellow()
is_car
yes
no
red()