The led display is a 7-segment display with four digits and a colon. It is able to display time and date.

In order to interact with the display, two libraries are included: *SevenSegmentTM1637*.h and *SevenSegmentExtended.h*.

The display is connected to pin 4 and 5, respectively CLOCK and DIO.

The whole behaviour of the display is basically managed by an interrupt. Every 0.5 sec the method displayTime() is executed. This method contains a switch to define the behaviour of the display in every machine state.   
In IDLS and RING state the 7-segments shows either the current time or the alarm time, according to the pressure of the black button. Moreover, displaying time the colon will blink thanks to the variable int status.

void displayTime() {

 static byte status = 0; //used for blinking, if it's 0 - the selected segments are off, when 1 the segments are on

 switch (state) {

   // in IDLE state, we either show the current time or the alarm time, depending on the showAlarm flag

   // in current time view, the semicolon is blinking using the status variable

   case IDLS:

   case RING:

     if (showAlarm) {

       display.printTime(alarmArray[0], alarmArray[1], true);

     } else {

       display.printTime(now.hour(), now.minute(), status);

     }

     break;

[...]

}

In hours editing states (EAHR, EHRS) the 7-segments shows respectively the time or the alarm time, blinking the first two digits. As example, here is the code of the EAHR state:

[...]

   case EAHR:

     display.printTime(alarmArray[0], alarmArray[1], true);

     if (status && !noBlink) {

       display.printRaw((uint8\_t)OFF, 0);

       display.printRaw((uint8\_t)OFF, 1);

     }

     break;

[...]

As the code shows, in this case both status and noblink variable are used to perform digit blinking. The status variable is commuted every time at the end of the function, while the noblink function is modified in the loop. The variable blinkTimer holds the time of the last blinking, while the variable curr represent the current moment: if the difference among these two values is more than 1000, it must blink again.

[...]

 if (blinkTimer + 1000 < curr){

   noBlink = 0;

 }

[...]

Status EAMI and EMIN for time and alarm minutes editing work like the previous states, but in these cases the third and forth digits are blinking instead.

