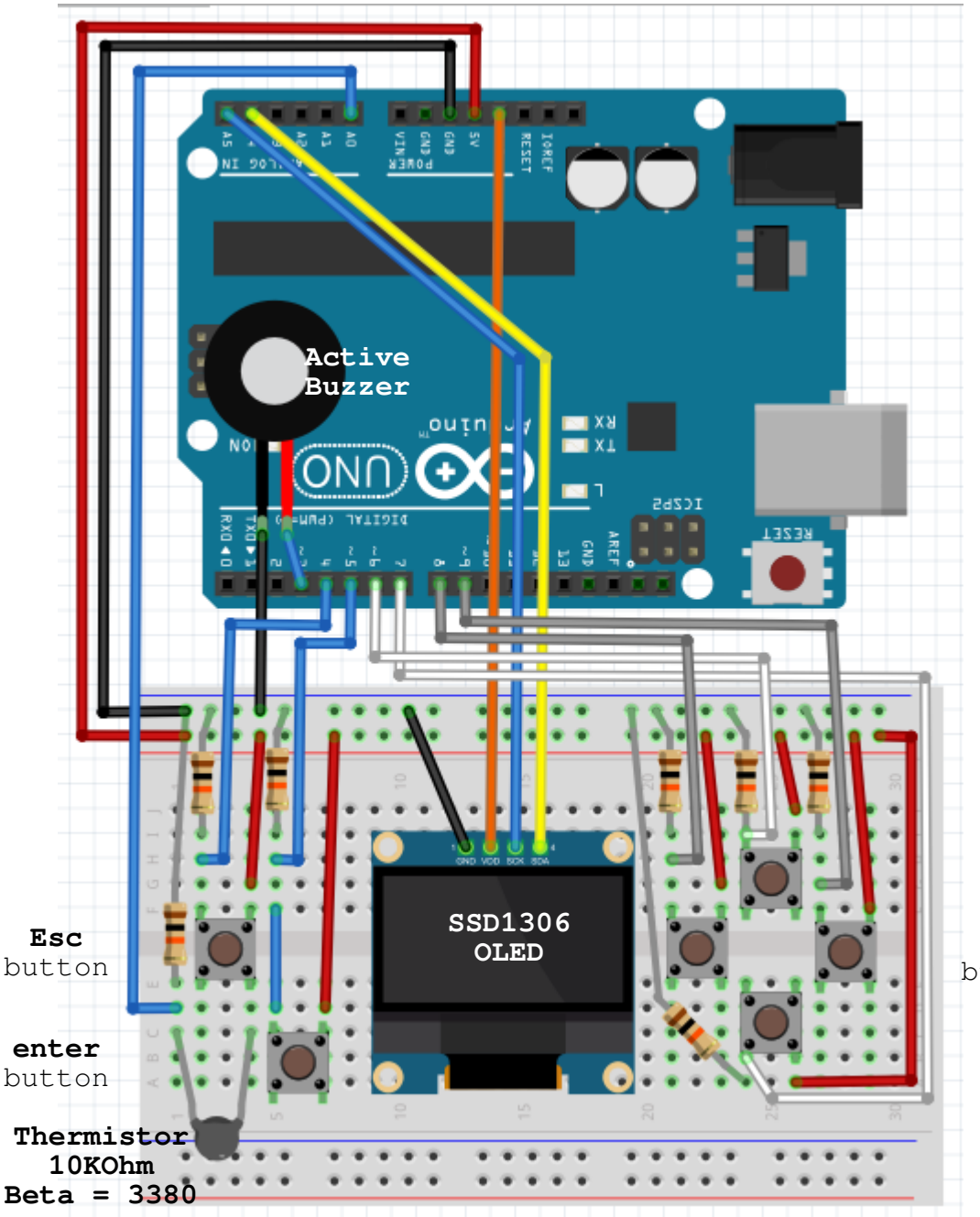
ATMEGA328/ARDUINO - PROJECT - DIGITAL ALARM CLOCK - OLED
https://github.com/teaksoon/p_daco



HARDWARE
1x Computer with Arduino IDE Software
1x USB 2.0 Type A/B Data Cable
1x Arduino Uno Board
1x Solderless Breadboard
Nx Jumper wires

1x Active Buzzer
6x Tactile Switch with 6x 10KOhm Resistor
1x SSD1306 OLED Module i2c 64x128 pixel

- Modular Design Extension -

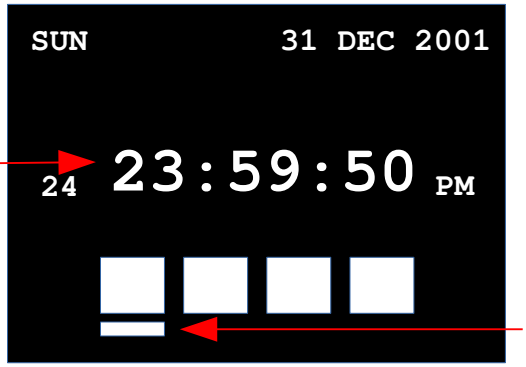
1x 10Kohm Thermistor(Beta=3380) with 1x 10KOhm Resistor

ATMEGA328/ARDUINO - PROJECT - DIGITAL ALARM CLOCK - OLED
https://github.com/teaksoon/p_daco

Source code: **p_daco_clock_ticker**
Download from:
https://github.com/teaksoon/p_daco/blob/main/2022_01_07_p_daco_source.zip
Upload PROGRAM, watch the OLED Screen

While in the MENU MODE and the Navigation Bar is at the CLOCK FUNCTION, the OLED screen will display CLOCK FUNCTION "Live" Clock (updated every 1 second)

Clock Ticker - "Live" Clock



ESC Button - Move Navigation Bar to first MENU OPTION

LEFT/RIGHT Button - Move MENU Navigation Bar. Show different FUNCTION Live Data

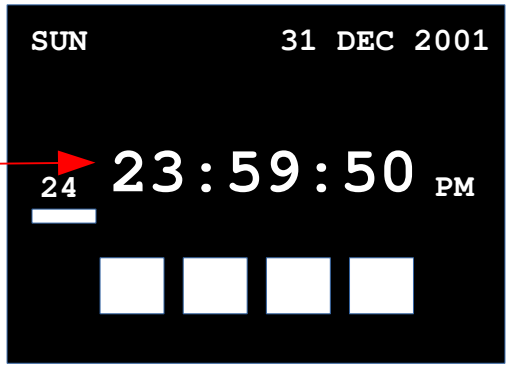
ENTER Button - Move into FUNCTION EDIT MODE

Presssing ENTER Button while in MENU MODE, will enter the EDIT MODE (CLOCK FUNCTION)

While in CLOCK FUNCTION EDIT MODE, "Live" Clock runs in the background, while the Screen allow Data Editing at Navigation Bar position

Clock Ticker - "Live" Clock stopped on the Screen for Data editing

- "Live" Clock is still running at the background, will resume when return from ESC



ESC Button - Cancel Edited Data and return to MENU MODE, continue with the previous Data

LEFT/RIGHT Button - Move Navigation Bar

UP/DOWN Button - Change data at the Navigation Bar position (yet to be coded)

ENTER Button - Save Edited Data and return to MENU MODE, continue from Saved Data (yet to be coded)

At this stage we can only display "Live" Time and Date. No Time/Date Editing yet. As you can see from the source codes in the next page, the coding just for the Time/Date ticker (increment by 1 seconds) is already quite alot codes (we handle both 12/24 hours mode).

Next, we will be coding the Time/Date Editing with the Buttons, more coding at that stage of coding because we need to swap between 12hours and 24hours mode and also swapping between AM/PM while editing hour and modes

Once that is done, "Live" Time/Date ticker Display with Time/Date Editing, we will have a fully functional running Clock

Code for Time/Date Ticker

```
void clock_update() {
char tmp[3];
  clo_ss++;  // seconds increased by 1
  if (clo_ss > 59) {
    clo_ss = 0;
    clo_mm++; // minutes increased by 1
    if (clo_mm > 59) {
      clo_mm = 0;
      clo_hh++; // hour increased by 1
      if (clo_hhmode == HHMODE_24) {
        // --- 24 HOURS MODE ---
        if (clo_hh == 24) {
          clo_ampm  = AM; // previously 23 hour, PM turns to AM, next day
          clo_hh    = 0;  // hour=24 is changed to 0
          clock_update_nextday();
        } else {
          if (clo_hh == 12) {
            clo_ampm = PM; // AM turns to PM at 12 hour, same day
          }
        }
      } else {
        // --- 12 HOURS MODE ---
        if (clo_hh == 13) {
          // Previously 12 AM/PM, we cannot have 13 in 12 hours mode
          // 13 is changed to 1, still same AM/PM
          clo_hh = 1;
        } else {
          if (clo_hh == 12) {
            // Previously was 11 AM/PM
            if (clo_ampm == PM) {
              // previously was 11PM, PM turns to AM, 12AM next day
              clo_ampm = AM;
              clock_update_nextday();
            } else {
              // previously was 11AM, AM turns to PM, 12PM same day
              clo_ampm = PM;
            }
          }
        }
      }
    }
  }
}
void clock_update_nextday() {
  clo_dow++;
  if(clo_dow > 7) clo_dow = 1; // cycle between 1 to 7

  clo_dd++; // day increased by 1
  if (clo_dd > mo_end[clo_mo-1]) {
    clo_dd = 1; // day exceeds month end, set to 1
    clo_mo++;   // month increased by 1
    if (clo_mo > 12) {
      clo_mo = 1; // month exceeds 12, set to 1
      clo_yyyy++; // yyyy increased by 1
      if (clo_yyyy > 9999) {
        clo_yyyy = 2001; // Maximum Year = 9999. Go back to default, 2001
      }
      mo_end[1] = (g_isLeapYear(clo_yyyy)==1) ? 29:28;
    }
  }
}
```