СИСТЕМНА АРХИТЕКТУРА

# Структурна диаграма

# 

Структурата се състои от 3 основни модула и 3 спомагателни модула. В Main модула се съдържа main функцията на програмата, както и безкрайния цикъл, който е сърцето на софтуера. В LCD Display модула се намира кода (функции), свързан с управлението на показваната информация. В Temperature Sensor модула се намира кода (функции и константи), служещи за измерването на температурата и нейното прочитане. Всеки от тези 3 модула си има по 1 спомагателен модул, съдържащ конфигурации, включително дефиниране на глобални константи и избор на пинове.

# Поведенческа диаграма

# 

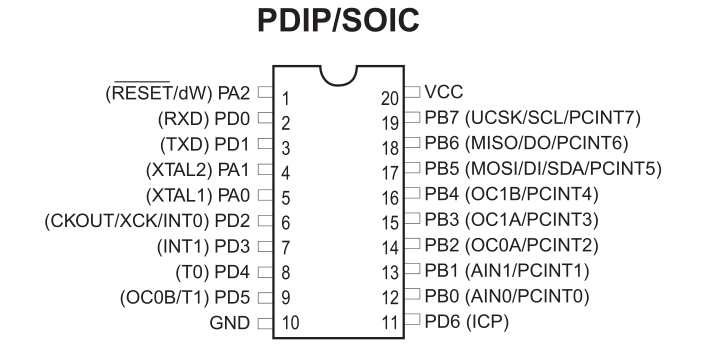
При стартиране на продукта, първоначално се правят инициализации, след което се влиза в безкрайния цикъл, докато към устойството е включено захранването. Действията, които се изпълняват в цикъла, включват измерване на температурата и определяне на желаната максимална и минимална температура. След което на база сравнение между желаната и текущата температура, се определя дали да се включи отоплението или охлаждането, или и двете да не работят, ако температурата е в исканите граници. След което се изпълнява програмно закъснение, преди изпълнението на същите стъпки отново.

# Компоненти

* **Процесор: ATtiny2313**
  + AVR – RISC архитектура.
  + 32 x 8 регистъра за общо предназначение
  + 2KB вградена програмируема флаш памет
  + 128 байта вградена програмируема EEPROM памет
  + 128 байта вградена SRAM памет
  + 20 пина
  + работно напрежение 2.7-5.5V
* **Температурен сензор DS18B20**
  + 1-wire интерфейс, изискващ само един пин за комуникация
  + измерва температурата от -55°C до +125°C
  + точност ±0.5°C при околна температура между -10°C и +85°C
  + време необходимо за измерване на температурата: 750ms
  + работно напреение 3.0-5.5V
* **LCD дисплей TC1604A-01 (16x4)**
  + 16x4 – 16 символа на ред, 4 реда
  + SPLC780D1 или еквивалентен контролер
  + 8-битов паралелен интерфейс
  + работно напрежение – 4.5-5.5V
* **Реле за веригата за охлаждане**
* **Реле за веригата за отопление**
* **Потенциометър за промяна на яркостта на диспея**

# Описание на портовете

1. RESET. Свързан с бутон за рестартиране на програмата.
2. PD0. Изходящ сигнал към релето, управляващо веригата за отопление.
3. PD1. Изходящ сигнал към релето, управляващо веригата за охлаждане.
4. PA1(XTAL2). Използван за връзка към тактов генератор (честота 16MHz).
5. PA0(XTAL1). Използван за връзка към тактов генератор (честота 16MHz).
6. PD2. Свободен.
7. PD3. Свързан към пин 4 на LCD дисплея – RS – определя дали се предават данни или инструкции.
8. PD4. Свързан към пин 5 на LCD дисплея – R/W – определя дали се чете или пише.
9. PD5. Свързан към пин 6 на LCD дисплея – E – chip enable сигнал.
10. GND. Земя.
11. PD6. Изпраща управляващ сигнал към температурния сензор.
12. PB0. Свързан към пин 7 на LCD дисплея – DB0 – шина за данни/инструкции.
13. PB1. Свързан към пин 8 на LCD дисплея – DB1 – шина за данни/инструкции.
14. PB2. Свързан към пин 9 на LCD дисплея – DB2 – шина за данни/инструкции.
15. PB3. Свързан към пин 10 на LCD дисплея – DB3 – шина за данни/инструкции.
16. PB4. Свързан към пин 11 на LCD дисплея – DB4 – шина за данни/инструкции.
17. PB5 (MOSI). Свързан към пин 12 на LCD дисплея – DB5 – шина за данни/инструкции. Свързан и към програматора.
18. PB6 (MISO). Свързан към пин 13 на LCD дисплея – DB6 – шина за данни/инструкции. Свързан и към програматора.
19. PB7 (UCSK). Свързан към пин 14 на LCD дисплея – DB7 – шина за данни/инструкции. Свързан и към програматора.
20. VCC. Захранване.



# Абстракции

## Температурен сензор DS18B20

Възможно е използването на opensource driver за датчика за температура. Примерен такъв код е достъпен на адрес

<http://svn.ohwr.org/white-rabbit/trunk/software/watchdog/arm_rev2/mini_backplane/dev/> .

DS18X20\_start\_meas(uint8\_t with\_power\_extern, uint8\_t id[]) : uint8\_t – Стартира измерване на температурата, чрез подаване на сигнал до сензора. Първия параметър избира дали се ползва външно захранване на пин 3, или паразитно (по шината за данни) на пин 2. Вторият параметър избира дали командата е предназна за сензор с определен сериен номер, или към първия намерен сензор на шината (при стойност NULL).

DS18X20\_read\_decicelsius(uint8\_t id[], int16\_t \*decicelsius) : uint8\_t – Функция за намиране на температурата. Първия параметър определя кой е модела на устойството (0x28 означава, че модела е DS18B20), във втория параметър се формира температурата в градуса децицелзий. Функцията връща число, имащо значение на код за грешка, или код за успех.

DS18X20\_format\_from\_decicelsius( int16\_t decicelsius, char str[], uint8\_t n) : uint8\_t – Функция за форматиране на низ от символи, показващи температурата. По подаден първи параметър температура (27.4°C = 274), инициализира втория параметър като форматиран низ от символи ( 27.4°C = +27.4), последния параметър опрделя дължината на символния низ, като при недостатъчна дължина се връща грешка. Функцията връща число, имащо значение на код за грешка, или код за успех.

Датчика работи при околна температура от -55.0°C до +125.0°C. При температура под минималната, или над максималната за сензора е необходимо на екрана да се изписва съобщение за грешка. Същото важи и при връщане на грешка по друга причина, например прекъсване на веригата (датчика не е свързан към останалите компоненти).

## LCD дисплей

lcd\_clear() : void – Подава инстукция за изчистване на дисплея.

lcd\_setcursor(uint8\_t x, uint8\_t y) : void – Задава позиция на курсора, според зададените координати x и y.

lcd\_home() : void – Задава начална позиция на курсора (координати: 1,1)

lcd\_data(uint8\_t data) : void - Показва на дисплея ASCII символа с код подадения параметър.

lcd\_string(const char\* data) : void – Показва на диспея символния низ, подаден като параметър.

# Файлова структура

Requirement.docx – Файл, съдържащ изискванията към проекта

Architecture.docx – Файл, съдържащ архитектурата на проекта.

Tests.docx – Файл, описващ проведените тестове и резултатите им

README.md – Файл, представящ проекта и екипа, работил над него

LICENSE – Файл, съдържащ лиценза на софтура

src – Директория, съдържаща програмния код

lcd – Директория, в която са поместени файловете, свързани с LCD дисплея

lcd.c – дефиниция на функции, използвани от и за дисплея

lcd.h – декларация на функции, използвани от и за дисплея

lcd\_config.h – конфигурация на пиновете и константи, свързани с дисплея

temp\_sensor – Директория, в която са поместени файловете, свързани с температурния сензор.

temp\_sensor.c – дефиниция на функции, използвани от и за сензора

temp\_sensor.h – декларация на функции, използвани от и за сензора

temp\_sensor\_config.h – конфигурация на пиновете и константи, свързани със сензора.

main.c – съдържа main функцията.

pins.h – дефиниции на портовете и пиновете.