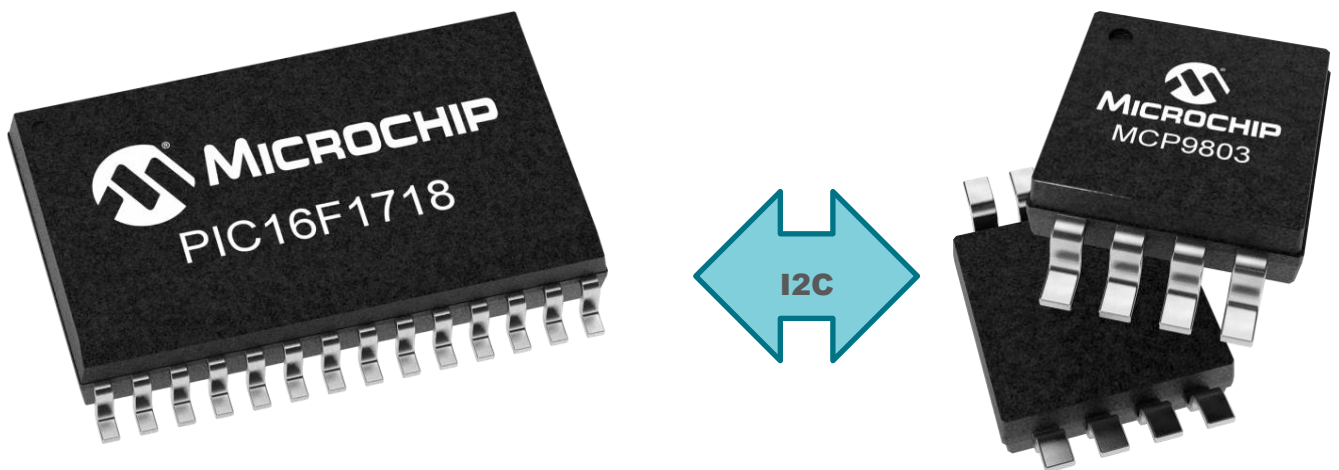


Построение системы контроля датчика температуры



Часть 1. Создание модуля опроса датчика температуры посредством микроконтроллера

Задача

Необходимо построить систему опроса и контроля датчика температуры с интервалом опроса 1сек., и передаче данных через последовательный порт (RS-232).

Выбор элементов

Построение проведен на микроконтроллере и датчике от компании MICROCHIP.

PIC16F1718 сочетает в себе интеллектуальную аналоговую интеграцию с низкой стоимостью и экстремально низкой мощностью (XLP) для различных приложений общего назначения. Этот микроконтроллер предоставляет встроенные операционные усилители, независимую от ядра периферию (CLC, COG, NCO и Zero Cross Detect) и выбор периферийных контактов, обеспечивая повышенную гибкость конструкции.

МСР9803 - цифровой датчик температуры, способный считывать температуру от -55°С до +125°С. Температурные данные измеряются с помощью встроенного датчика температуры и преобразуются в цифровое слово с помощью выбираемого пользователем 9-12-битного Сигма-Дельта ADC преобразователя

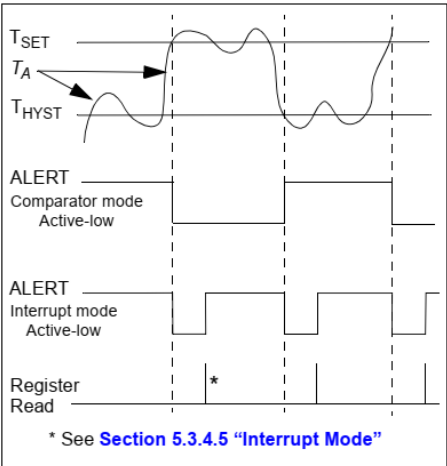
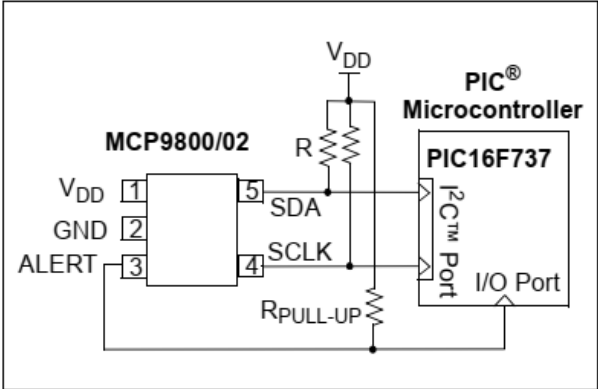
<p>PIC16F1718 - 8-bit Flash-микроконтроллер</p>	<p>МСР9800/1/2/3 - цифровой датчик температуры</p>
<p><u>Особенности:</u></p> <ul style="list-style-type: none"> 28 KB Flash Program Memory, 2 KB Data SRAM 128 B of Non-volatile Data Storage 25 I/O Pins Four 8-bit Timers / One 16-bit Timer 2 x High-Speed Comparators 2 x Op Amps 17ch x 10-bit ADC One 8-bit DAC and One 5-bit DAC Zero Cross Detect 2 x CCP / 2 x PWM 1 x COG, 4 x CLC, 1 x NCO 1 x EUSART, 1 x I2C/SPI 	<p><u>Особенности:</u></p> <ul style="list-style-type: none"> Преобразователь температуры в цифровой Точность с 12-битным разрешением: <ul style="list-style-type: none"> - $\pm 0,5^{\circ}\text{C}$ (типичный) при температуре $+25^{\circ}\text{C}$ - $\pm 1^{\circ}\text{C}$ (максимум) от -10°C до $+85^{\circ}\text{C}$ - $\pm 2^{\circ}\text{C}$ (максимум) от -10°C до $+125^{\circ}\text{C}$ - $\pm 3^{\circ}\text{C}$ (максимум) от -55°C до $+125^{\circ}\text{C}$ Выбираемое пользователем разрешение: 9-12 бит Диапазон рабочего напряжения: от 2,7 В до 5,5 В 2-проводный интерфейс: Совместимость с I2C/SMBus Рабочий ток: 200 мкА (типичный) Ток отключения: 1 мкА (максимум)

MCP9803

МСР9800/1/2/3 имеет четыре регистра, доступных пользователю. Эти регистры определены как: регистр температуры окружающей среды (TA), регистр предельного набора температура (TSET), температурный гистерезис (THYST) регистр и конфигурация устройства (CONFIG). Также имеется ALERT pin - выходной pin с открытым коллектором. Устройство выдает предупреждающий сигнал, когда температура окружающей среды выходит за пределы запрограммированного пользователем температурного предела.

В режиме *Comparator* выход ALERT устанавливается, когда TA больше TSET. Сигнал остается активным до тех пор, пока TA не станет ниже THYST. Режим компаратора полезен для применений типа термостата, таких как включение охлаждающего вентилятора или отключение системы, когда температура превышает безопасный рабочий диапазон.

Typical Application



В режиме компаратора, если устройство переходит в режим выключения с заявленным сигнальным выходом, выход остается активным во время выключения. Устройство должно работать в режиме непрерывного преобразования (*continuous conversion*), с TA ниже THYST, чтобы сигнал ALERT был отключен.

Адресные биты A2, A1, A0 являются входными контактами устройства или ведомого адреса и доступны только с MCP9801/03. Адреса устройств MCP9800/02 установлены на заводе-изготовителе.

MCP9803 driver

Библиотека для драйвера состоит из следующих составных частей:

<ul style="list-style-type: none"> MCP9803.h MCP9803.c 	Определяет класс объекта датчика, значения состояния датчика и функции класса
<ul style="list-style-type: none"> MCP9803_interface.h MCP9803_interface.c 	Определяет интерфейс взаимодействия с датчиком по I2C (на основании стандартного интерфейса)
<ul style="list-style-type: none"> MCP9803_conversion.h MCP9803_conversion.c 	Определяет преобразование данных, полученных от датчика в формате two's complement format в: <ul style="list-style-type: none"> float (для дальнейших вычислений) char* (для передачи в последовательный порт)

Основная часть программы содержит конфигурационный файл с выбором режимов работы:

<ul style="list-style-type: none"> MCP9803_init.h MCP9803_init.c 	<ul style="list-style-type: none"> Инициализирует датчик MCP9803, определяем режимы работы датчика (continuous/one_shot), определяет функции прерываний (ALARM,READY)
---	--

MCP9803 initialized values

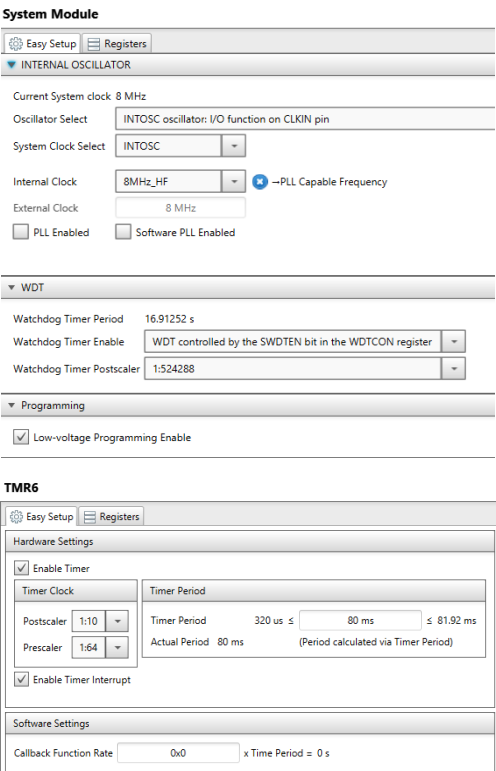
Исходя из имеющихся характеристик и входных требований, определим значения следующих параметров:

<ul style="list-style-type: none"> ADC resolution 	10bit
<ul style="list-style-type: none"> ADC Conversion Time 	2*(30ms...75ms) = 80ms
<ul style="list-style-type: none"> Period of Getting Temperature Value 	2000ms = 2sec
<ul style="list-style-type: none"> SLEEPING time period, автопробуждение по WDT 	WDT >16sec
<ul style="list-style-type: none"> FAULT_QUEUE (длина очереди ложного срабатывания) 	1 (default)
<ul style="list-style-type: none"> WakeUp по сигналу ALERT, не уходить в сон пока сигнал установлен 	необходим "Continuous Mode" необходим "ComparatorALERT Mode"
<ul style="list-style-type: none"> По предустановке на одном из пинов микроконтроллера предусмотреть работу в одном из режимов "Continuous Mode" / "One-Shot Mode" 	Jumper setting

PIC16F1718

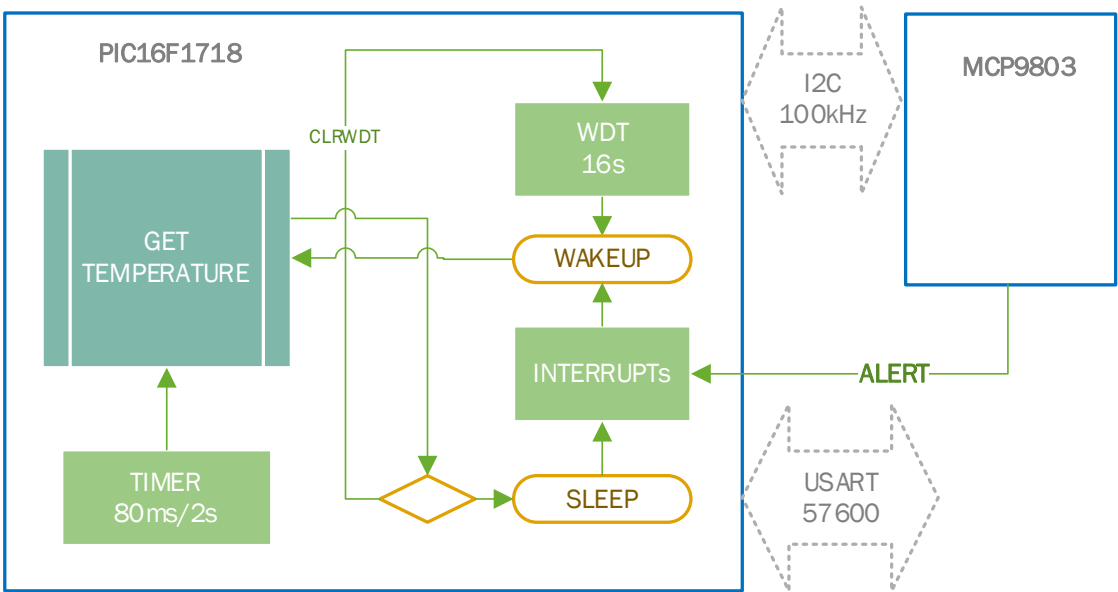
Для управления данным датчиком MCP9803, от микроконтроллера потребуется настройка (MPLAB X IDE) следующих блоков и элементов периферии:

OSCILLATOR MODULE	INTOSC – Internal oscillator (8MHz no PLL)
WATCHDOG TIMER	WDT is controlled by software
I2C	MSSP I2C Master 100kHz Interrupt enable
USART	Baudrate 57600 Only transmit
PIN	RB4 – OneShotMode_PresetPin RC3 – I2C SCL RC4 – I2C SDA RC5 – ALERT (IOC Interrupt any edge) RC6 – USART Tx
TMR6	Postscaler – 1:10 Prescaler – 1:64 TimerPeriod = 80ms = MCP9803_ADC_Delay Enable Timer Interrupt TMRCallback_CounterMAX=25 => Tperiod = 2s



Блок-схема взаимодействия

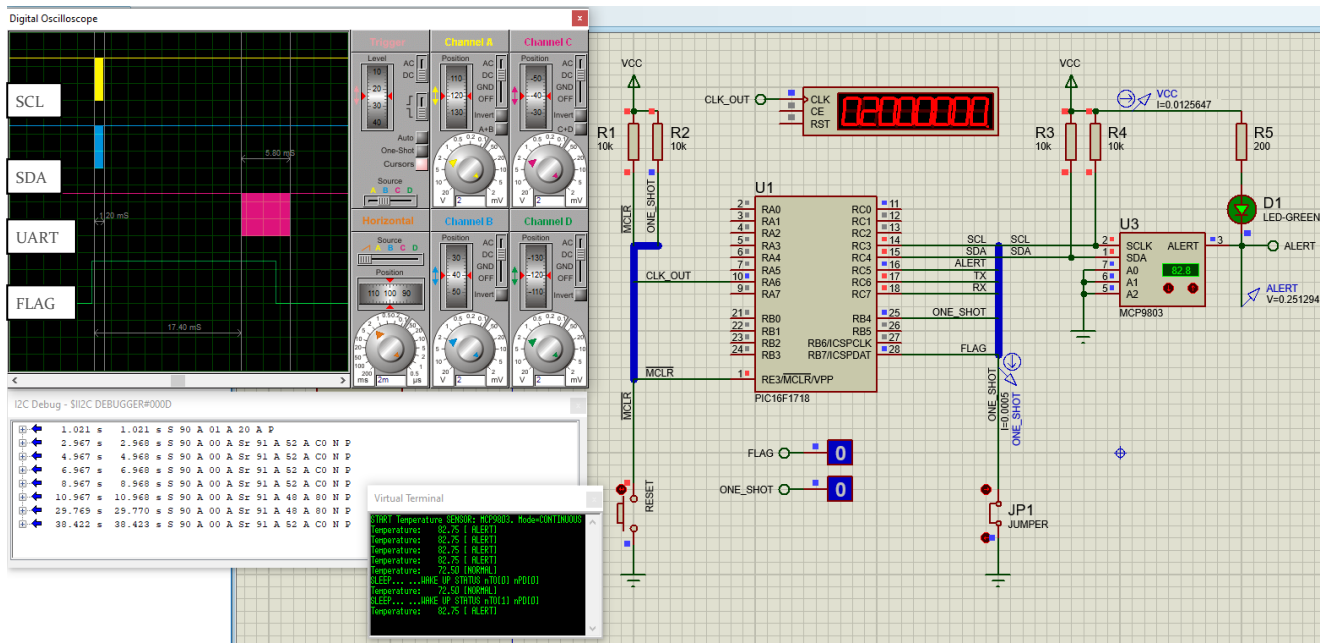
Исходя из имеющихся характеристик и входных требований, сформируем обобщенную схему взаимодействия основных модулей микроконтроллера с датчиком и прочими элементами:



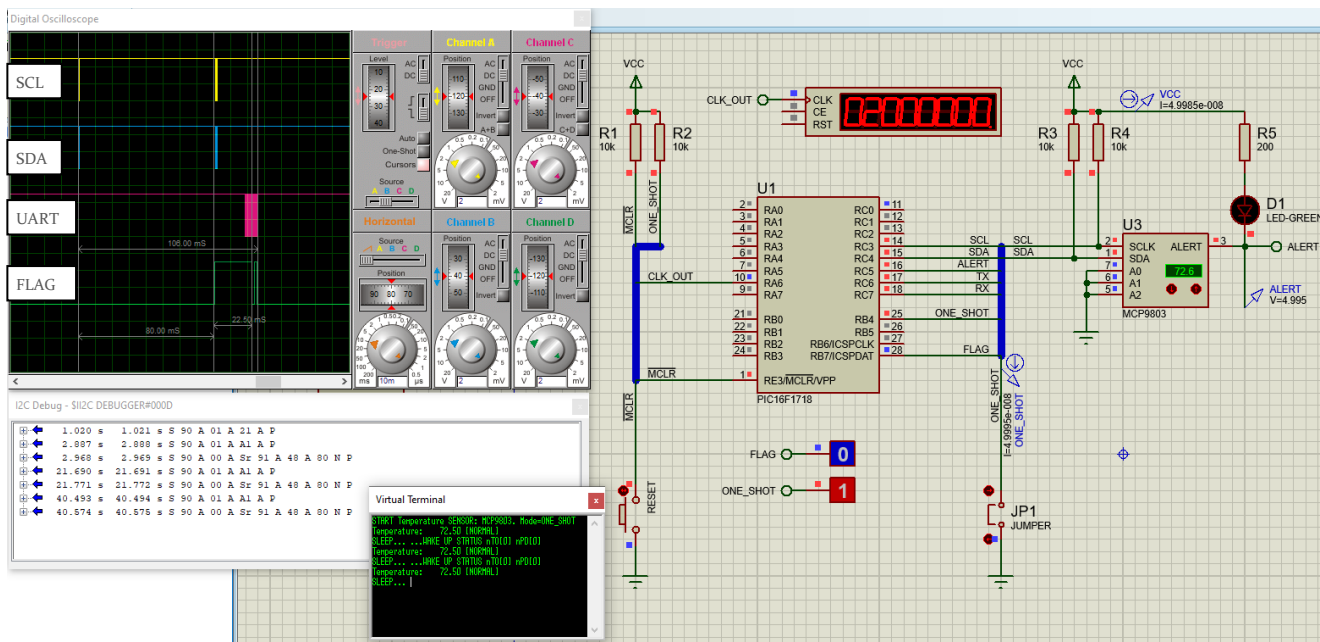
Моделирование в PROTEUS

Исходя из имеющихся характеристик и входных требований, сформируем обобщенную схему взаимодействия основных модулей микроконтроллера с датчиком и прочими элементами.

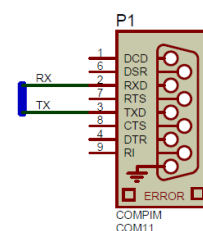
1. "Continuous Mode"



2. "One-Shot Mode"



В конечном итоге полученную модель с моделированием можно подключить к COM-порту (например, виртуальному COM11) и отслеживать через него данные с датчика.



Часть 2.

Определение формата задания параметров и считывания данных

Параметры датчиков хранятся в конфигурационном файле. Разделители значений параметров и пример приведены ниже:

sensor_parameters.txt

```
#SENSOR'S PARAMETERS
#
#GROUPNAME_TYPES: <!!! not able to change !!!>>
#      NAME
#      VALUE
#INDICATOR_TYPES:
#      LCD      #indicator LCD
#      FLG      #indicator of FLAGS
#SEPARATOR_TYPES:
#      GROUPS_SEPARATOR
#      G NAMES_SEPARATOR
#      INDICS_SEPARATOR
#      PARAMS_SEPARATOR
#      VALUES_SEPARATOR
#
#EXAMPLE TEMPLATE =====
#NAME: Name;
#VALUE: {INDICATOR} [MeasureName Units MinScaleValue ... .. MaxScaleValue]; #comments
#=====
#
#GROUPS_SEPARATOR:      [;]\s*
#G NAMES_SEPARATOR:     [;]\s*
#INDICS_SEPARATOR:      [{}]\s*
#PARAMS_SEPARATOR:      [N\]\s*
#VALUES_SEPARATOR:      [ ]\s*
#
#=====

NAME: MCP9800; VALUE: {LCD} [Temperature °C -50 125]; VALUE: {FLG} [Alarm LOW HIGH]

NAME: MCP9803;
VALUE: {LCD} [Temperature °C -50 75 80 125]
VALUE: {FLG} [Alarm NORMAL MIDDLE ALERT FATALITY]
```

Аналогично, относительно применяемых разделителей данных, считываем значения с датчиков:

sensor_testMessage.txt

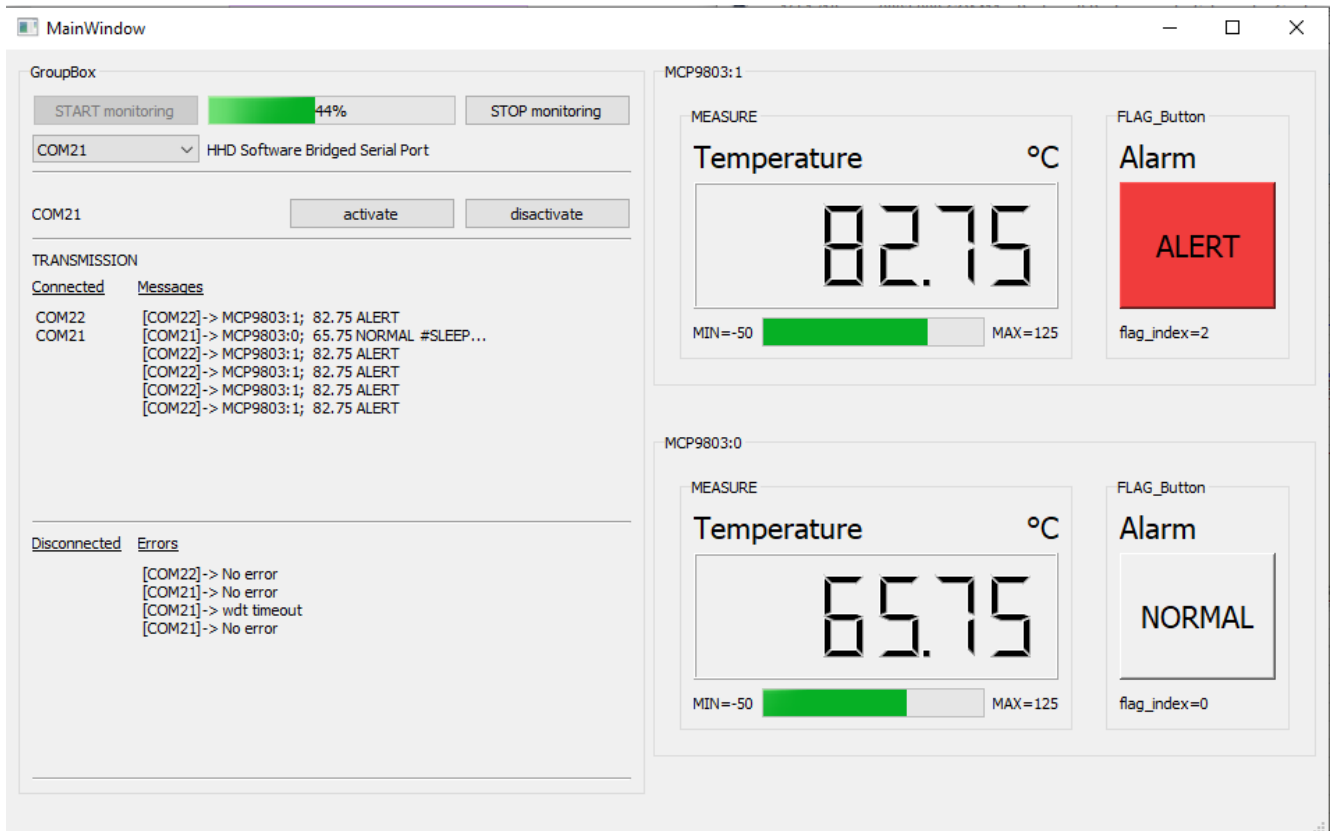
```
#TEST Messages
#
MCP9803:001; 55      NORMAL      #SLEEP...
MCP9801:001; 62.3    NORMAL
MCP9803:001; 66.9    NORMAL
MCP9803:001; 78.0    NORMAL
MCP9803:001; 82.1    ALERT
MCP9803:001; 81.9    ALERT
MCP9803:001; 75.2    ALERT
MCP9803:001; 69.8    NORMAL
MCP9803:001; 50.1    NORMAL
MCP9803:001; 50.4    NORMAL
#
```

Часть 3.

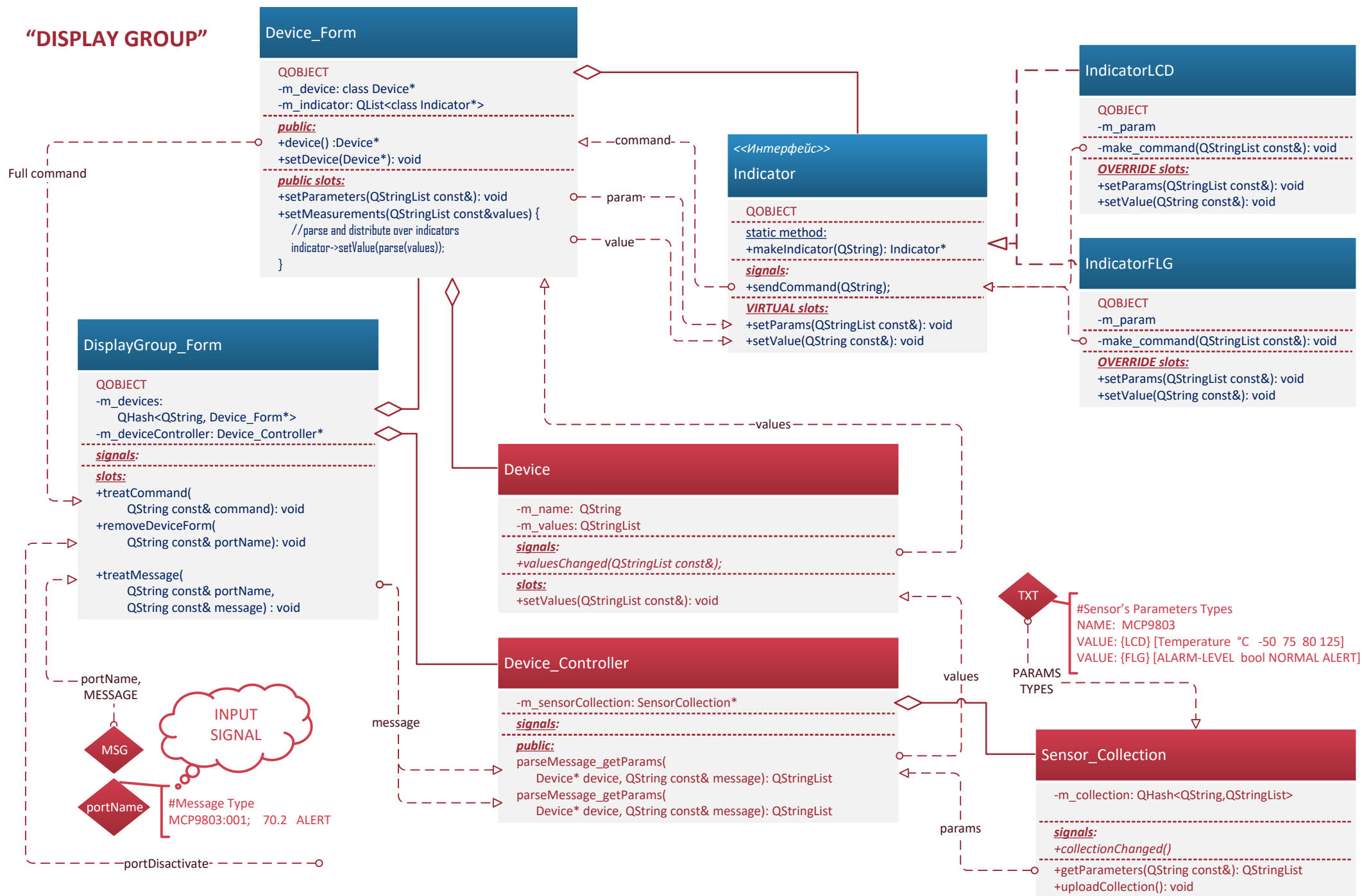
Описание программы мониторинга

1. Мониторинг портов производится с заранее выбранной периодичностью (20сек). Из списка детектированных портов пользователь имеет возможность выбрать и активировать выбранные им порты, которые будут переведены в разряд «активированных» и по ним будет производиться обработка данных.
2. Если по определенному порту будет отсутствовать передача данных в течении заранее определенного промежутка времени(30сек), выбранный порт будет деактивирован. Порт также будет деактивирован в случае, если за указанный промежуток времени прочитанные данные не будут соответствовать ни одному из датчиков, внесенных в конфигурационный файл.
3. В случае удачной расшифровки данных, полученных по каждому из открытых портов от соответствующих датчиков, на основании параметров, указанных в конфигурационном файле, на экране появятся индикаторы. В случае деактивирования порта, соответствующие индикаторы будут закрыты.
4. Программа написана на C++ в Qt IDE с учетом следующих особенностей:
 - a. Опрос открытых com-портов производится в отдельном потоке. Созданный объект порта типа QSerialPort в случае удовлетворения всех параметров будет перемещен в этот поток и из него формировать и отсылать данные.
 - b. Для определения времени ожидания данных создается таймер типа QTimer, аналогичный таймеру “watchdog”, который сбрасывается каждый раз по приходу новых данных. В случае истечения времени таймера выдается сигнал на закрытие порта.
 - c. Класс индикаторов строится на принципах ООП: в основе находится базовый класс, определяющий общий интерфейс, конкретные классы индикаторов наследуются от базового. Также базовый класс имеет статический метод формирования конкретного класса индикатора, исходя из буквенной аббревиатуры (factory pattern).
 - d. В интерфейсе индикаторов предусмотрена возможность реагирования пользователя на пришедшую визуальную информацию о состоянии датчика либо о численных показаниях: возможно испускание команд.
 - e. Главное окно mainWindow_Form разбито на две части:
 - i. отвечающая за коммуникацию с портами – ComPortGroup_Form, и
 - ii. отображающие данные с датчиков – DisplayGroup_Form.
 - f. Основной принцип взаимодействия между классами основан на сигнально-слотовых соединениях.

Общая схема взаимодействия классов приведена ниже в приложении. В приведенном ниже изображении экрана монитора видно, что имеется 2 активных com-порта COM21 и COM22, которые передают данные от датчиков температуры MCP9803: MCP9803:000 и MCP9803:001 соответственно. Для обоих определено отображение данных температуры на “IndicatorLCD”(LCD) и данных состояния на “IndicatorFLAG”(FLG).



"DISPLAY GROUP"



“COMFORT GROUP”

