

# Цифровой термометр на микроконтроллере MSP430P325 с автономным питанием

Предлагаемый цифровой термометр предназначен для измерения температуры в широком диапазоне от  $-50$  до  $+500^{\circ}\text{C}$ . В качестве датчика температуры используется платиновый датчик PT1000, который монтируется в выносном зонде и подключается к термометру при помощи разъема типа BNC (приборный разъем CP-50). Такое решение позволяет применять с термометром несколько зондов, имеющих различную конструкцию: контактную, погружаемую, воздушную и т. д. Реальная температура, измеряемая термометром, определяется конструктивным оформлением зонда.

Термометр может служить для измерения температуры в самых различных областях: в пищевой промышленности, медицине, нефтяной промышленности и др. Там, где необходимо контролировать температурный режим с высокой точностью, цифровой термометр будет хорошим помощником. Основные особенности описываемого термометра заключаются в длительном сроке работы без замены батареи питания и малой погрешности измерения температуры. Это стало возможным благодаря применению в нем микроконтроллера MSP430P325 компании Texas Instruments. В термометре реализованы часы реального времени (Real Time Clock, RTC), фиксация измеренного значения температуры и режим "Auto Off".

## Конструкция цифрового термометра

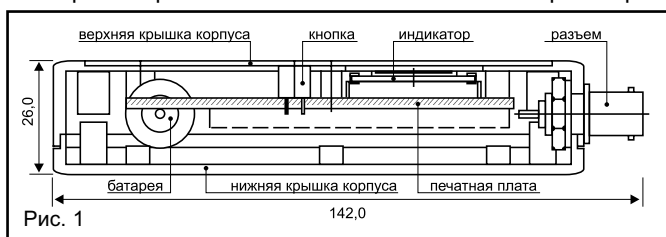
Термометр смонтирован на двухсторонней печатной плате. На ее лицевой стороне помещены ЖК-индикатор и кнопки. Микроконтроллер и остальные элементы установлены на обратной стороне. В цифровом термометре применены SMD-компоненты. Печатная плата термометра помещена в стандартный корпус, в верхней части которого находится разъем для температурного зонда. Расположение основных элементов цифрового термометра внутри корпуса показано на рис. 1. Печатная плата крепится четырьмя винтами к бобышкам на верхней крышке корпуса. Последняя закрыта специальной пленкой, на которую методом шелкографии нанесены надписи. Индикатор защищен окном из прозрачной пластмассы толщиной 0,6 мм. Крышки термометра крепятся друг к другу четырьмя винтами. Для достижения герметичности корпуса все швы перед сборкой проклеиваются силиконом.

Термометр обладает следующими характеристиками:

- погрешность измерения (аппаратная) — в диапазоне от  $-50$  до  $+199,9^{\circ}\text{C}$  —  $\pm 0,1^{\circ}\text{C}$ ;
- погрешность измерения (аппаратная) — в диапазоне от  $+200$  до  $+450^{\circ}\text{C}$  —  $\pm 1^{\circ}\text{C}$ ;
- период измерения температуры равен 1 с;
- точность внутренних часов составляет  $\pm 10$  с в сутки;
- индикация разряда батареи;
- $4\frac{1}{2}$ -разрядный ЖК-индикатор с высотой цифр 9 мм;
- габариты: 65x130x25 мм;
- вес 120 г;
- питание от литиевой батареи 3,6 В;
- срок службы батареи при работе 12 часов в сутки — не менее 5 лет.

## Принципиальная схема цифрового термометра

На рис. 2 приведена принципиальная электрическая схема термометра. Базовый элемент схемы — микроконтрол-



лер MSP430P325 — идеальная микросхема для создания энергонезависимых измерительных приборов. Микроконтроллер MSP430E325 с EPROM-памятью программ используется для отработки программного обеспечения, а однократно программируемый микроконтроллер MSP430P325 — в серийной продукции. Микроконтроллер MSP430P325 имеет встроенный стабилизатор тока 0...6 мА и 12-разрядный АЦП с расширением до 14 разрядов. Поэтому измерение температуры с применением микроконтроллера MSP430P325 не представляет особых трудностей.

Резистор R2 задает ток, протекающий через датчик PT1000. Конденсаторы C1, C3 и резистор R3 представляют собой простейший фильтр. Остальные элементы необходимы для нормального функционирования прибора. Кнопки SB1–SB3 управляют режимами работы термометра. Резисторы R4–R6, R12 и конденсатор C8 используются схемой обслуживания кнопок. Резистор R7 и микросхема DA1 применяются для формирования опорного напряжения в программе контроля разряда батареи питания. Резисторы R8–R11 определяют режим работы встроенного драйвера жидкокристаллического индикатора HL1. ЖК-индикатор типа 5066 имеет три подложки и работает в мультиплексном режиме.

Литиевая батарея Bat1 обеспечивает прибор питанием. Микроконтроллер MSP430P325 (DD1) управляет всеми перечисленными элементами. Благодаря его использованию, количество компонентов в схеме минимально. Основные элементы, необходимые для работы термометра — АЦП, стабилизатор тока, драйвер ЖК-индикатора, аналоговый мультиплексор, таймеры — встроены в микроконтроллер.

Кнопка SB1 (On/Off) включает/выключает термометр. Кнопка SB2 (Hold/Set) в режиме "температура" фиксирует показания измеренной температуры на индикаторе или включает режим непрерывного измерения. Кнопка SB3 (C/Time) переключает режим работы термометра "температура / часы".

## Алгоритм измерения температуры

Сопротивление датчика PT1000 зависит от внешней температуры. При температуре  $20^{\circ}\text{C}$  сопротивление датчика составляет 1000 Ом, отсюда и тип датчика — PT1000. Зависимость сопротивления датчика PT1000 от температуры описывается формулой:

$$R(T) = 1000 \cdot (1 + 3,908 \cdot 10^{-3} \cdot T - 5,802 \cdot 10^{-7} \cdot T^2),$$

где R — сопротивление датчика, T — температура. Температурный градиент датчика равен  $3,9 \text{ Ом}/^{\circ}\text{C}$ . Для того, чтобы вывести на индикатор значение текущей температуры, необходимо измерить сопротивление датчика PT1000. Эта задача решается аппаратным и программным путем. Программное обеспечение микроконтроллера MSP430P325 включает в себя библиотеку математики с плавающей точкой (Floating Point Package), поэтому проблем с вычислениями быть не должно.

Библиотека оперирует 24-разрядными числами, точность вычислений более чем достаточна. Внимательно взглянув

на приведенную выше формулу, вы увидите, что это обыкновенное квадратное уравнение. Как вычислить корень квадратного уравнения, известно из курса алгебры для 7 класса. Напомню:  $Y=AX^2+BX+C$ , где  $Y$  соответствует  $R$  – сопротивления датчика (Ом), а  $X$  – температуре (°C). Преобразуя формулу зависимости сопротивления датчика от температуры к виду  $Y=AX^2+BX+C$ , получим формулу:

$$R(T) = -5,802 \cdot 10^{-4} \cdot T^2 + 3,908 \cdot T + 1000.$$

Естественный вопрос: откуда взять значение  $R$ ? Резистор  $R2$  задает ток, протекающий через датчик. Ток вычисляется по формуле:

$$I = 0,249 \cdot V_{cc}/R2,$$

где  $R2 = 1,5$  кОм,  $V_{cc} = 3,6$  В (напряжение батареи Bat1). Отсюда  $I = 0,5976$  мА.

Микроконтроллер, используя АЦП, измеряет падение напряжения на сопротивлении датчика по входу А0. Измеренное напряжение равно произведению числа, полученного от АЦП, на вес младшего разряда АЦП:  $U = N \cdot 0,2197$  мВ. Делим значение напряжения, полученное при измерении, на известный ток через датчик и получаем сопротивление датчика на момент измерения. Ток через датчик является постоянной величиной, поэтому после всех преобразований получим формулу, по которой программа и вычислит значение температуры:

$$T = \frac{-3,908 + \sqrt{17,593 - 0,0008535 \cdot N}}{-0,0011604},$$

где  $N$  – число, полученное в результате работы АЦП,  $T$  – значение температуры в градусах Цельсия, выводимое на индикацию.

Фрагмент программы, вычисляющий значение температуры, показан ниже. Все вычисления производят с использованием библиотеки FPP.

```

adc_cm_1  call  #FLT_SAV      ;Сохраняем значения регистров в
стеке
sub  #4,sp
;Результат работы АЦП сохраняем в ячейках с начальным адресом "res".
mov  r9,res
; Преобразуем результат АЦП в формат FPP и сохраним результат преоб-
разования в ячейках Nx.
mov  #res,RPARG
call  #CNV_BIN16
mov  @sp+,Nx
mov  @sp,Nx+2
sub  #2,sp
;Вычисляем значения температуры по приведенной выше формуле.
mov  #Nx,RPRES
mov  #c_08_1,RPARG
call  #FLT_MUL              ;N Ч 0.0008532
jmp  adc_cm_n
;.....
adc_cm_n  mov  #c_1759,RPRES ;
call  #FLT_SUB              ;17.593 - (N Ч 0.0008532)
call  #FLT_SQRT            ;Извлечение квадратного корня
mov  #c_3908,RPARG
call  #FLT_ADD              ; -3.908 + предыдущий результат
mov  #c_0116,RPARG
call  #FLT_DIV              ;Предыдущий результат / -0.0011604
mov  #const6,RPRES
call  #FLT_MUL              ;Предыдущий результат Ч 1000000
mov  sp,RPARG
; Преобразовываем результат вычисления в BCD формат для вывода на ин-
дикацию.
call  #CNV_FP_BCD
mov  @sp+,res
mov  @sp,res+2
add  #4,sp
call  #FLT_REC              ; Восстанавливаем значения регистров.
; В ячейках памяти с начальным адресом "res" хранится результат вычис-
лений в формате BCD: res 220-221h 00 00 (+), 80 00 (-), 222-223h 05 32(053.2)

```

### Алгоритм работы программы

Основной принцип работы программы состоит в том, что микроконтроллер должен максимально долго находиться в "спящем" режиме. Благодаря этому достигается минимальное энергопотребление. Микроконтроллер не должен находиться в циклах "опроса". Поэтому все события обрабатываются по прерываниям, позволяющим исключить непроизводительные ожидания событий в циклах. Пятнадцать векторов прерывания микроконтроллера MSP430P325 позволяют программисту эффективно использовать рабочее время микроконтроллера.

Основной таймер микроконтроллера (Basic Timer) программируется на генерацию вектора прерывания с периодом 500 мс. При генерации вектора прерывания вызывается программа "Int\_BT", обрабатывающая это событие:

Основной таймер микроконтроллера (Basic Timer) программируется на генерацию вектора прерывания с периодом 500 мс. При генерации вектора прерывания вызывается программа "Int\_BT", обрабатывающая это событие:

```

Int_BT  mov
#SSEL+CNTCL+WDTPW,&WDTCTL ; Запуск
Watchdog.

```

В случае зависания программы сторожевой таймер микроконтроллера (Watchdog) перезапустит систему, восстановив работу цифрового термометра.

```

push  r10
push  r11
inv.b  &SEC_05
cmp.b  #0fh,&SEC_05
;По содержимому ячейки SEC_05 равным 0fh
вызов подпрограммы "ЧАСЫ" - 1 раз в секунду.
jz  int_btct
call  #clock              ;Вызов подпрограммы
"ЧАСЫ"(+1 sec).
int_btct  cmp.b  #00h,&ON_OFF ;Тер-
мометр включен?
Jz  int_bt0              ;Да, работаем!

```

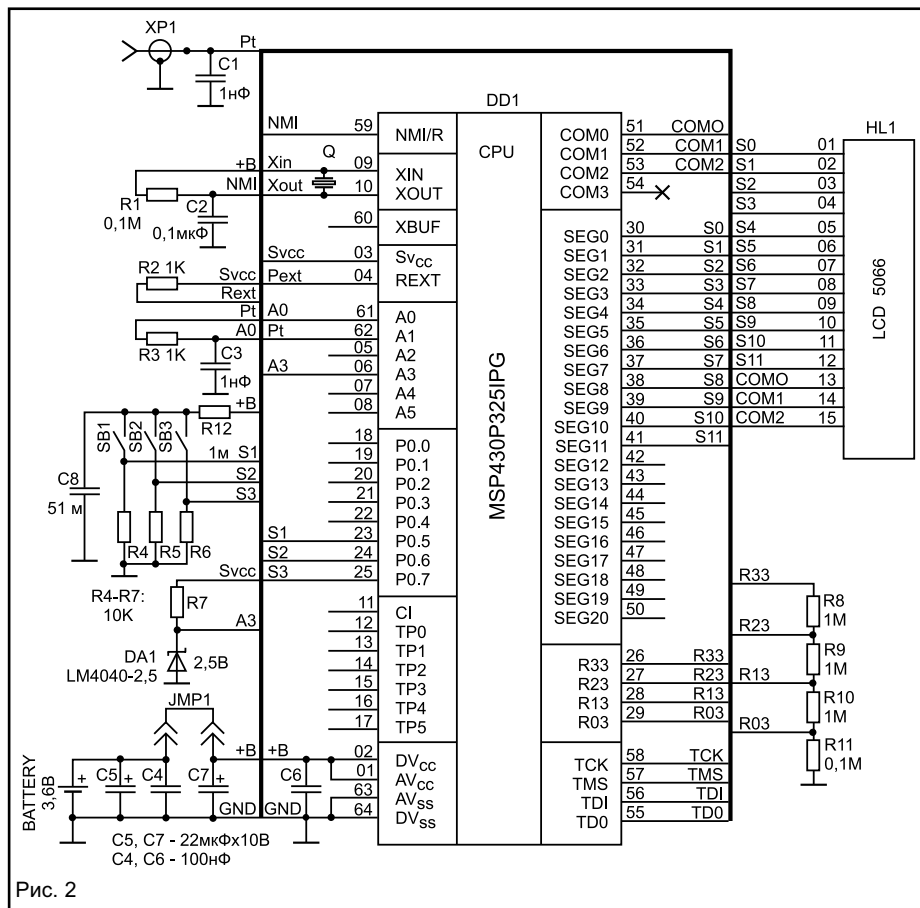


Рис. 2

```

Pop r11 ;Нет, "спать"!
Pop r10
Reti
;.....
int_bt0 mov.b LB_buf,L_B
        cmp.b #00,&MODE ;Режим: температура
или часы?
        jz int_bt2 ;Режим: температура.
        jmp int_clk ;Режим: часы.

int_bt2 cmp.b #0ffh,SEC_05
        jz int_bt_mrg ;Индикация
        jmp int_temp ;Измерение + индикация
    
```

Временная диаграмма работы цифрового термометра показана на рис. 3. Основное время микроконтроллер находится в "спящем" режиме, и благодаря этому достигается потрясающая характеристика энергопотребления термометра. Средний ток, потребляемый цифровым термометром от батареи, не превышает 30 мкА.

Цикл работы цифрового термометра начинается с подачи импульса тока через RT1000, заданного резистором R2 (в генераторе тока), длительностью 1 мс. В АЦП, встроенный в микроконтроллер, включена схема выборки/запоминания. Поэтому специальных мер по стабилизации входного напряжения применять не надо. Время запоминания входного напряжения составляет 30 мкс, время преобразования – 60 мкс. После измерения падения напряжения на датчике RT1000, выключается генератор тока, и данные обрабатываются программой с использованием библиотеки с плавающей точкой.

Для повышения стабильности показаний используется следующий прием. Результаты измерения хранятся в четырех последовательных ячейках памяти Nx1–Nx4. После каждого измерения информация в ячейках Nx1–Nx4 сдвигается вправо, новая информация записывается в ячейку Nx1. Программа вычисляет среднее значение четырех результатов измерения. Результат усреднения пересчитывается в градусы Цельсия и выводится на индикатор. Активное время работы программы обработки – 9 мс. После обработки результатов измерения микроконтроллер переводится в "спящий" режим. Через 500 мс микроконтроллер "просыпается", выполняет программы "Часы" и "Индикация" и снова переходит в "спящий" режим. Через 500 мс цикл повторяется. Отношение рабочего режима микроконтроллера к "спящему" составляет 1/100 (10 мс/1000 мс). При этом средний ток, потребляемый цифровым термометром в рабочем режиме, не превышает 40 мкА (при работе 24 часа в сутки). Для дополнительного снижения энергопотребления введен режим "Auto Off". По истечении времени, заданного режимом "On/Off" (1...8 часов), прибор автоматически переходит в "спящий" режим с выключением индикации. Повторное включение возможно кнопкой "On/Off". Малое время протекания тока через датчик делает использование этого прибора привлекательным для измерения низких температур.

Входы P0.5–P0.7 микроконтроллера использованы для обслуживания кнопок. Аппаратное решение клавиатуры также преследует цель снижения энергопотребления. В исходном состоянии конденсатор C8 заряжен через резистор R12 сопротивлением 1 МОм. При нажатии любой кнопки вызывается процедура обработки вектора прерывания порта P0. Вызывается подпрограмма обслуживания клавиатуры. При удержании кнопки в нажатом положении ток через замкнутую кнопку

не превышает 3 мкА. После отпускания кнопки конденсатор C8 заряжается за 50 мс и клавиатура снова готова к работе.

Фрагмент программы, обслуживающей функцию "Часы", приведен ниже. Программа корректирует показания часов (плюс 1 секунда) и одновременно контролирует режим "Auto Off". В "спящем" режиме функция "Auto Off" отключена.

```

;----- подпрограмма Clock (Часы) -----
Clock clrc ;очистка признака переноса
dadd.b #1,CLK_S ;+1 в ячейке секунд
cmp.b #60h,CLK_S ;равно 60 секунд?
Jlo clock_exs ;нет
clr.b CLK_S ;да, обнулیم секунды
clrc
dadd.b #1,CLK_M ;+1 в ячейке минут
cmp.b #60h,CLK_M ;равно 60 минут?
Jlo clock_exm ;нет, переход на clock_exm
clr.b CLK_M ;да, обнулیم минуты
clrc
dadd.b #1,CLK_H ;+1 в ячейке часы
cmp.b #24h,CLK_H ;равно 24 часа?
Jlo clock_exh ;нет, выход
clr.b CLK_H ;да, обнулیم часы
jmp clock_exh
clock_exs ret
;Контроль режима "AutoOFF" и коррекция показаний индикатора ЖК
clock_exm mov.b CLK_M,LCDBUF_L ;коррекция индикации
cmp.b #0ffh,&ON_OFF ;спящий режим?
Jnz cti_off ;нет
clr.b &OFF_MIN ;да, обнулیم ячейки контроля режима
"AutoOFF"
clr.b &OFF_HOUR ;
jmp cti_off1 ;выход
cti_off1 inc.b OFF_MIN ;+1 минута в ячейке контроля
cmp.b #60,OFF_MIN ;равно 60 минут?
Jlo cti_off1 ;нет
inc.b OFF_HOUR ;да, +1 час в ячейке контроля
cmp.b &OFF_HOUR,&OFF_CTL ;выключить термометр?
Jnz cti_off1 ;нет
clr.b &OFF_MIN ;да, обнулیم ячейку минут
clr.b &OFF_HOUR ;да, обнулیم ячейку часов
mov.b #0ffh,&ON_OFF ;да, установим признак выключения
mov.b #0ffh,&LCDBUF_M
mov.b #0ffh,LCDBUF_L
call #LCD_DSP
bic.b #01,&LCDCTL ;да, выключим индикацию ЖК
ret ;выход
clock_exh mov.b CLK_M,LCDBUF_L ;коррекция индикации минут
mov.b CLK_H,LCDBUF_M ;коррекция индикации часов
cti_off1 ret ;выход
    
```

### Управление цифровым термометром

Управление цифровым термометром осуществляется при помощи трех кнопок:

- кнопки "On/Off", которая служит для включения и выключения термометра. В выключенном состоянии ток потребления не превышает 1 мкА;
- кнопки "°C/Time", которая переключает термометр из режима "температура" в режим "часы" и обратно;
- кнопки "Hold/Set", которая фиксирует измеренное значение температуры. При этом показания температуры на индикаторе мигают. Повторное нажатие кнопки возвращает термометр в режим непрерывного измерения температуры.

В режиме "часы" первое нажатие кнопки "Hold/Set" переводит термометр в режим установки "Auto Off". На индикаторе появляется мигающая цифра от 1 до 8. Кнопкой "°C/Time" можно установить желаемое время автоматического выключения термометра. Второе нажатие на кнопку "Hold/Set" фиксирует установленное время "Auto Off" и переводит термометр в режим установки времени. На индикаторе мигает значение часов. Кнопкой "°C/Time" корректируют показания часов. По третьему нажатию кнопки "Hold/Set" корректируют показания минут при помощи кнопки "°C/Time" Четвертое нажатие кнопки "Hold/Set" возвращает термометр в режим "часы".

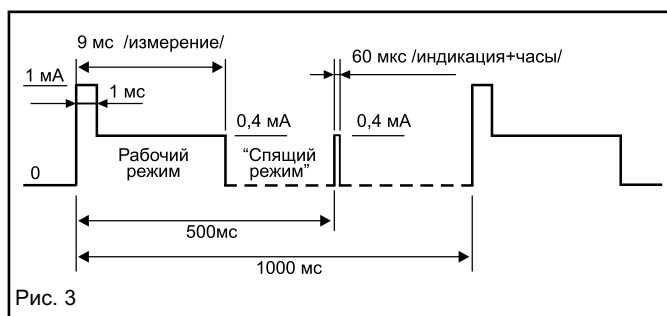


Рис. 3

Виктор Заикин,  
victor.zaikin@mail.ee