

Done at Home

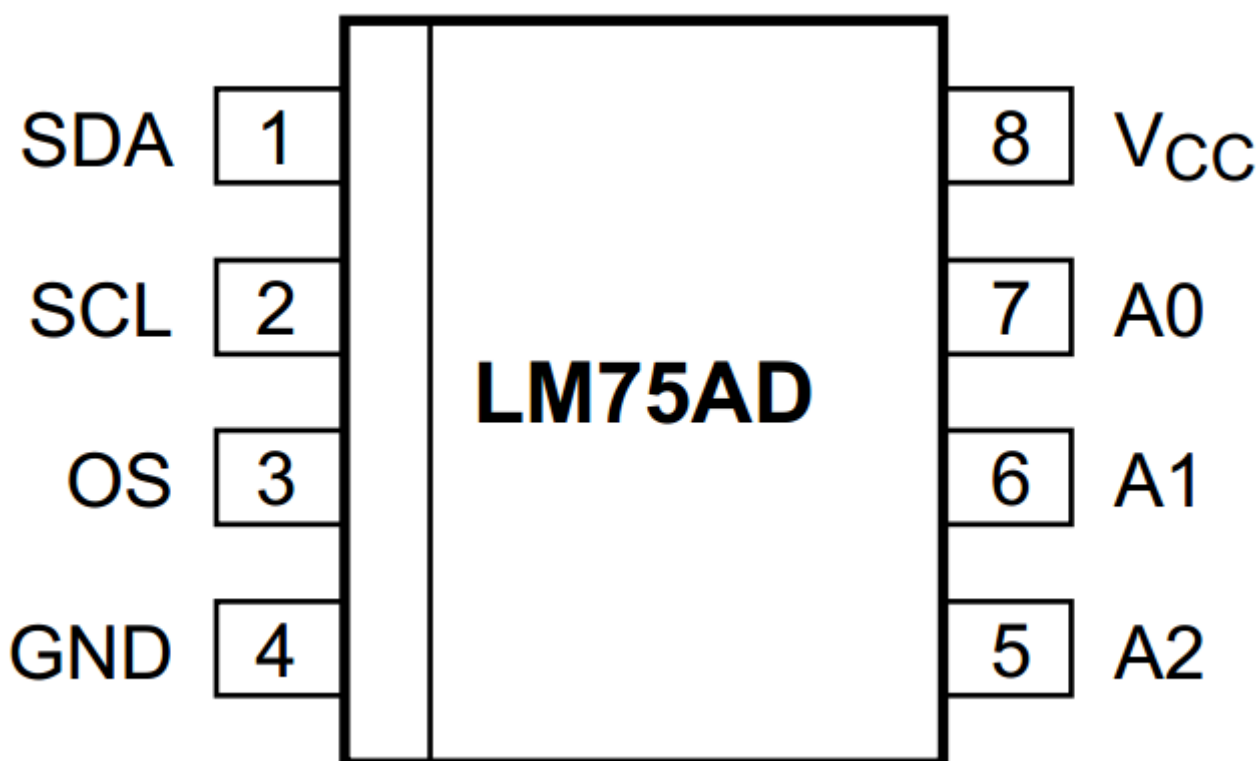
Программирование микроконтроллеров avr

Подключение датчика температуры LM75AD к (avr)

admin | 15.05.2014

0 Comment

Подключение датчика температуры LM75AD к (avr)



Статьи в помощь :

AVR микроконтроллеры для начинающих (урок 11) I2C(TWI)-интерфейс

AVR микроконтроллеры для начинающих (урок 12) UART/USART в avr

Подключение avr микроконтроллера к компьютеру

Некоторые особенности (ссылка)(документация):

- 1) Диапазон напряжения питания от 2,8 В до 5,5 В
- 2) I2C-интерфейс, до 8 устройств на одной шине
- 3) Измерение температуры в диапазоне от -55 °C до 125 °C
- 4) 11-разрядный АЦП, который предлагает температурное разрешение 0,125 °C
- 5) Автономная работа как термостат при включении питания

Описание выводов

1 – SDA (Serial Data Input/Output — вход/выход последовательных данных) – вывод входа/выхода для двухпроводного последовательного интерфейса. Вывод SDA – с открытым стоком и требует внешнего подтягивающего резистора.

2 – SCL (Serial Clock Input — вход последовательных синхроимпульсов) – используется для синхронизации данных по последовательному интерфейсу.

3 – OS Вывод для сигнала о перегрев, с открытым стоком.

4 – GND Земля.

5 – A2 Цифровой вход. Определяет адрес микросхемы: бит 2.

6 – A1 Цифровой вход. Определяет адрес микросхемы: бит 1.

7 – A0 Цифровой вход. Определяет адрес микросхемы: бит 0.

8 – Vcc Питание.

Общие сведения

Устройство может работать в одном из двух режимов: **normal** или **shutdown**. При работе в режиме **normal**, преобразование температуры выполнен каждые 100 мс и регистр Temp обновляется в конце каждого преобразования. В режиме **shutdown**, устройство переходит в режим ожидания, преобразования не происходят и регистр Temp держит последний результат; Однако интерфейс I2C по-прежнему активен и операции записи/чтения могут быть выполнены. Режим работы устройства определяется бит B0 регистра конфигурации. Кроме того, в конце каждого преобразования в нормальном режиме, регистр температуры (Temp) автоматически сравнивается с регистром, где хранится пороговое значение (Tos) + читается значение гистерезиса из регистра (Thyst) для изменения лог сост ножки OS.

Адрес микросхемы

Адрес микросхемы состоит из 2-ух частей: 1001 (не программируемая часть)+(программируемая часть)

MSB				LSB		
1	0	0	1	A2	A1	A0

Регистры

Имя регистра	Адрес регистра	Чтение/запись	значение по умолчанию	Описание
Temp	00h	read only	n/a	2 восьмибитных регистра для хранения значения температуры
Conf	01h	R/W	00h	1 восьмибитный регистр для хранения настроек
Thyst	02h	R/W	4B00h	2 восьмибитных регистра для хранения значения гистерезиса
Tos	03h	R/W	5000h	2 восьмибитных регистра для хранения значения температуры для выхода OS

Регистр указатель

Регистр Указатель содержит байт данных, из которого два младших бита это значение указателя на один из четырех регистров, а остальные биты всегда равны нулю. Регистр указателя не доступен

пользователю, но используется для выбора регистра данных для записи/чтения операцию включая байта данных указателя в посылку по I2C.

B7	B6	B5	B4	B3	B2	B[1:0]
0	0	0	0	0	0	pointer value

Table 7. Pointer value

B1	B0	Selected register
0	0	(Temp)
0	1	(Conf)
1	0	(Thyst)
1	1	(Tos)

Регистр температуры (Temp)

Это регистр состоящий из 2-ух байт хранит в себе значение измеренной температуры. Видно что для кодирования значения температуры используются 11бит. Так же приведена таблица температур и значений регистра температуры.

MSByte								LSByte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X

11-bit binary (2's complement)	Hexadecimal value	Decimal value	Value
011 1111 1000	3F8	1016	+127.000 °C
011 1111 0111	3F7	1015	+126.875 °C
011 1111 0001	3F1	1009	+126.125 °C
011 1110 1000	3E8	1000	+125.000 °C
000 1100 1000	0C8	200	+25.000 °C
000 0000 0001	001	1	+0.125 °C
000 0000 0000	000	0	0.000 °C
111 1111 1111	7FF	-1	-0.125 °C
111 0011 1000	738	-200	-25.000 °C
110 0100 1001	649	-439	-54.875 °C
110 0100 1000	648	-440	-55.000 °C

Регистры сравнения(Tos) и гистерезиса(Thyst)

Эти регистры одинаковы (9 бит). Tos- это регистр для сравнения измеренной температуры для определения состояния ножки OS.

Thust- это регистр для хранения гистерезиса (это значение нижнего порога для сброса ножки OS)

MSByte								LSByte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X	X	X

Table 13. Tos and Thyst limit data and value

11-bit binary (2's complement)	Hexadecimal value	Decimal value	Value
0 1111 1010	0FA	250	+125.0 °C
0 0011 0010	032	50	+25.0 °C
0 0000 0001	001	1	+0.5 °C
0 0000 0000	000	0	0.0 °C
1 1111 1111	1FF	-1	-0.5 °C
1 1100 1110	1CE	-50	-25.0 °C
1 1001 0010	192	-110	-55.0 °C

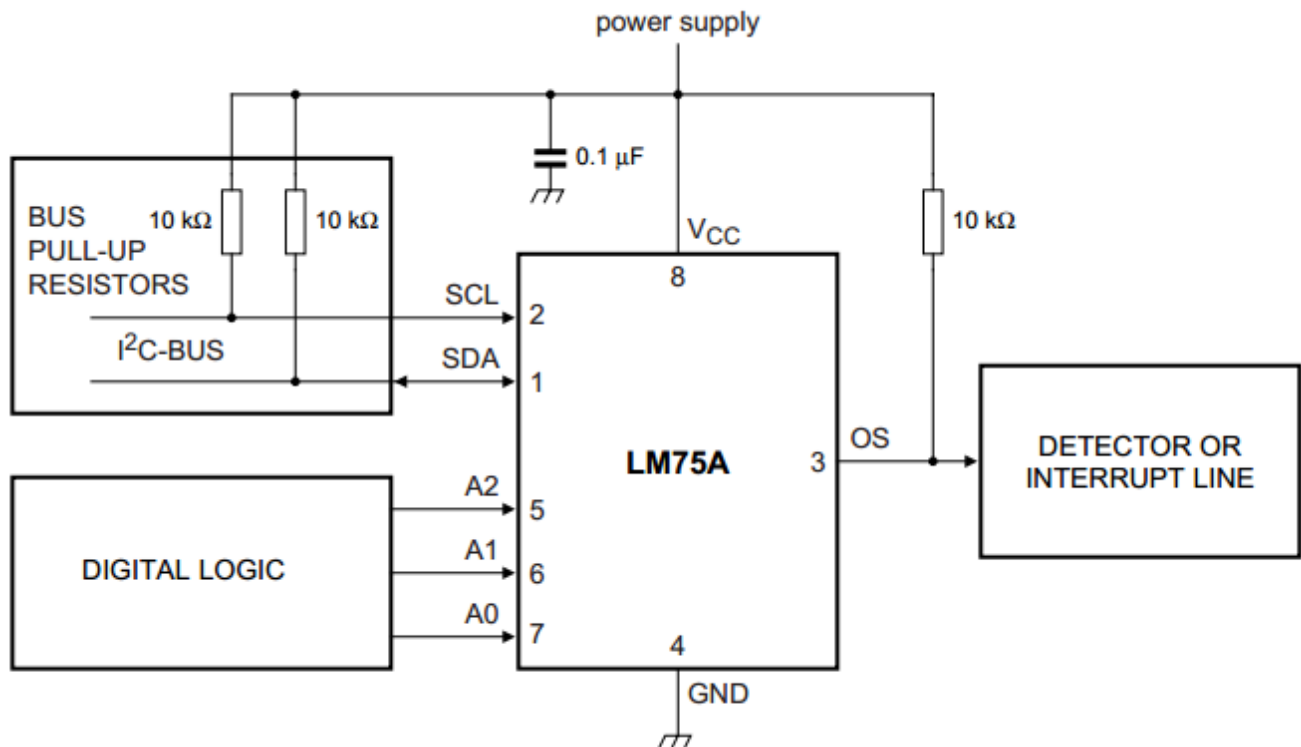
Регистр температуры (Conf)

Возможностей датчика не особо много, поэтому все настройки уместились в 1-ом регистре.

Значение с (*) - это значение по умолчанию

Бит	Имя	Доступ	Значение	Описание
B[7:5]	reserved	R/W	000*	Зарезервированы для использования изготовителем; должны быть нулями для нормальной работы
B[4:3]	OS_F_QUE[1:0]	R/W	OS fault queue programming	
			00*	queue value = 1
			01	queue value = 2
			10	queue value = 4
			11	queue value = 6
B2	OS_POL	R/W	OS polarity selection	
			0*	OS active LOW
			1	OS active HIGH
B1	OS_COMP_INT	R/W	OS operation mode selection	
			0*	OS comparator
			1	OS interrupt
B0	SHUTDOWN	R/W	device operation mode selection	
			0*	normal
			1	shutdown

Схема подключения



Ссылки на комплектующие:

Микроконтроллер: АТмега32 (ссылка)

LM75AD (ссылка)

Макетная плата: (ссылка)

Плата-переходник на dip (ссылка)

Конденсаторы: Рекомендую покупать наборы разных номиналов (ссылка)

Резисторы: Рекомендую покупать наборы разных номиналов (ссылка)

Ссылки на код и документацию:

Код: (LM75AD+TWI)+(документация)

LM75AD_TWI.h

```

1 #define LM75AD_ADRES 0b10010000 //адрес регистра микросхемы
2 #define TEMP 0 //адрес регистра Temp
3
4 //функция чтения температуры
5 // adr - адрес микросхемы LM75AD (0-7)
6 // возвращает 2байтную переменную LSB и MSB
7 unsigned int LM75AD_TWI_GetTemp (unsigned char adr)
8 {
9     unsigned int b;
10    unsigned int Temp;
11    I2C_StartCondition(); //генерируем условие СТАРТ
12    I2C_SendByte(LM75AD_ADRES|(adr<<1)); //адрес+бит записи
13    I2C_SendByte(TEMP); //отправляем адрес регистра Temp
14    I2C_StartCondition(); //генерируем условие СТАРТ
15    I2C_SendByte((LM75AD_ADRES|(adr<<1))|1); //адрес+бит чтения
16    Temp=I2C_RecieveByte(); //читаем 1-ый байт MSB
17    b=I2C_RecieveLastByte(); //читаем 2-ой байт LSB
18    I2C_StopCondition(); //генерируем условие СТОП
19    Temp=(Temp<<8)|b; //комбинируем int переменную из 2-ух байт MSB и LSB
20    return Temp;
21 }
22

```

```
23 //функция принимает значение регистра Temp и возвращает
24 //температуру от -55 до 127 шаг 1
25 char LM75AD_TWI_Convert (unsigned int TT)
26 {
27 return (TT>>8); //оставляем 8 бит
28 }
```

LM75AD.c

```
1 #define F_CPU 1000000UL
2 #include <avr/io.h>
3 #include <util/delay.h>
4 #include "TWI.h"
5 #include "LM75AD_TWI.h"
6
7 //макросы вычисления скорости
8 #define BAUD 1200
9 #define UBRR_VAL F_CPU/16/BAUD-1
10
11 void Uart_init (unsigned int speed)
12 {
13 // устанавливаем скорость Baud Rate: 1200
14 UBRRH=(unsigned char)(speed>>8);
15 UBRL=(unsigned char) speed;
16 UCSRA=0x00;
17 UCSRB|=(1<<TXEN)|(1<<RXEN); // Разрешение работы приемника
18 UCSRB|=(1<<RXCIE); // Разрешение прерываний по приему
19 // Установка формата посылки: 8 бит данных, 1 стоп-бит
20 UCSRC=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);
21 }
22
23 int main (void)
24 {
25 I2C_Init(); //инициализация модуля I2C
26 Uart_init(UBRR_VAL); // инициализация модуля
27
28 unsigned int y;
29
30 while(1)
31 {
32 _delay_ms(2000);
33 UDR=LM75AD_TWI_Convert((LM75AD_TWI_GetTemp(0)));
34 }
35 }
```

Подключение датчика температуры LM75AD к (avr) (...)



Подключение датчика температуры LM75AD к (avr) (...)



Рубрика: Все посты AVR Подключаем к AVR Статьи AVR

[сюда](#)