

# Замена DS2409 на 12F629

Андрей Шабронов (shabronov@ngs.ru)

Микросхема DS2409 – это специальная микросхема и ключевой элемент увеличения длины линии интерфейса 1-wire. Она выполняет коммутацию шин, что позволяет «суммировать» и «вычитать» электрическую нагрузку элементов сети, создавать таким способом различные топологии и подключать достаточно большое количество элементов 1-wire. Поскольку выпуск DS2409 прекращён, а потребность использования осталась, в данной статье приведено описание схемы и программы замены DS2409 на широкодоступный микроконтроллер 12F629. Возможно применение микроконтроллеров данной серии для замены других элементов интерфейса 1-wire. Отметим, что микроконтроллер даёт возможность добавить новые функции систем измерений и оставить старые элементы интерфейса. Сохраняется наработанное программное обеспечение информационных систем, устраняется монополизм производителей специализированных микросхем и расширяется функциональность систем телеметрии.

## Принципиальная схема

Принципиальная схема замены микросхемы DS2409 [1] на микроконтроллер (МК) 12F629 [2] для использования с интерфейсом 1-Wire фирмы Dallas Semiconductor представлена на рис. 1.

Схема выполняет «ветвление» или «разделение» шины 1-Wire на участки, поэтому используется термин «схема ветвителя» или «ветвитель». Стрелками показано состояние приёма или

передачи данных для соответствующего вывода МК.

Схема содержит узел управления на микросхеме U1 (12F629) и управляемые ключевые элементы на транзисторах Q1...Q5 (2N7000).

Диоды D1...D3 (1n5817) выполняют защитные функции от возможных импульсов положительных и отрицательных напряжений, наводимых на линию интерфейса 1-Wire. Кон-

денсатор C1 фильтрует возможные импульсные помехи по линии питания +5 В.

Поскольку интерфейс 1-Wire двунаправленный, для коммутации сигналов используется последовательное включение двух транзисторов от одного сигнала управления. Транзисторы Q 1, 2 управляются от вывода 6 (GP1) МК U1 (12F629), транзисторы Q 3, 4 управляются от вывода 3 (GP4) МК U1 (12F629).

Резисторы R1...R4 выполняют функцию «подтягивания» линии 1-Wire к положительному уровню +5 В. При установке МК на выводах управления высокого логического уровня +5 В управляемые транзисторы открываются, и линия разъёма XP3 (1w) соединяется с разъёмами XP1 (aux-1w) или XP2 (main-1w). Суммарное сопротивление соединения не превышает одного Ома.

И наоборот, при установке МК на выводах управления низкого логического уровня 0 В транзисторы закрываются, и суммарное сопротивление устанавливается максимально большим. Выходная линия отключается, и при этом выполняется «подтяжка» к уровню +5 В через резисторы R2, R3.

Контроль выходных шин MAIN-1W и AUX-1W на возможное короткое замыкание выполняется МК через вывод 2 (GP5) и вывод 5 (GP2). Таким образом, передавая сигнал запроса, можно определить исправность коммутируемых линий.

Управление МК осуществляется в стандарте 1-Wire. Через вывод 4 (GP3) сигналы принимаются от шины 1-Wire и через вывод 7 (GP0) передаются в линию 1-Wire.

Передачу сигнала выполняет управляемый транзистор Q5, который замыкает шину 1-Wire к уровню 0 В и тем самым формирует из него сигнал 0 стандарта 1-Wire для приёма на компьютере.

Временные интервалы замыкания формирует МК U1 (12F629). При использовании внутреннего RC генератора в 4 МГц погрешность сигналов управления составляет  $\pm 1$  мс, что не вызывает ошибок в сигналах при приёме и идентификации данных интерфейса 1-Wire.

Необходимо учесть дополнительный потребляемый ток МК в 1...3 мА по шине

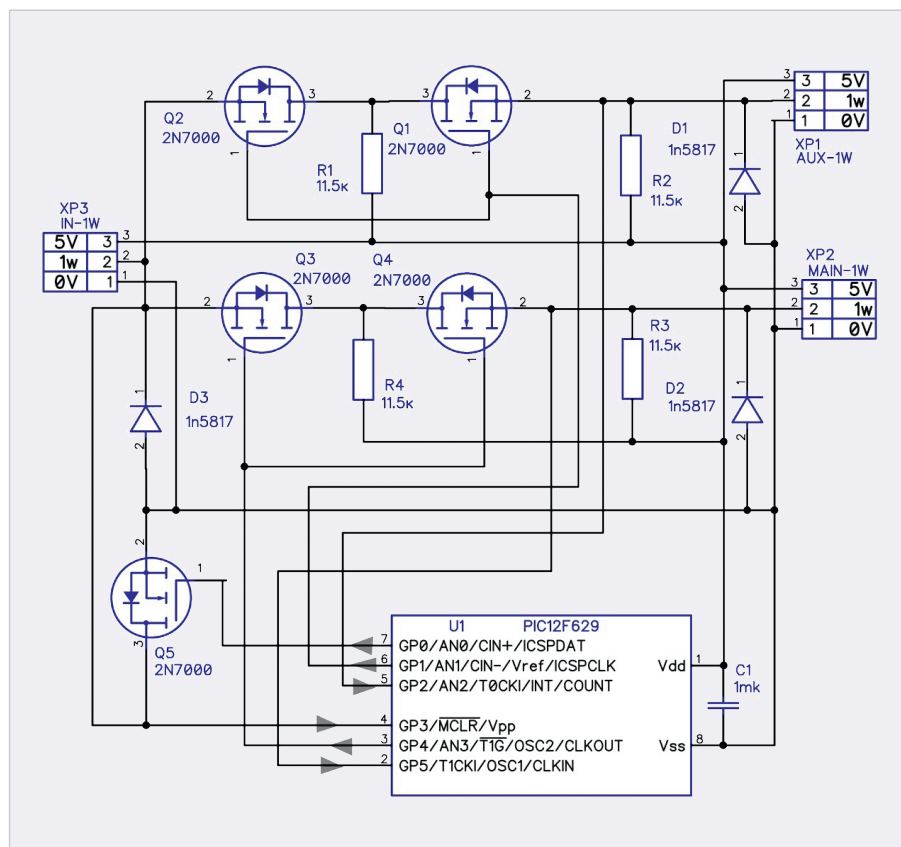


Рис. 1. Принципиальная схема замены DS2409 на 12F629

+5 В и возможное падение напряжения для длинной шины питания.

## Конструкция

Схема и все компоненты собраны на печатной плате и предназначены для монтажа под «винт». На рис. 2 представлены: слева фото собранной печатной платы, справа 3D-модель той же платы. Печатная плата разработана в среде проектирования DipTrace, проект доступен в каталоге программ [3] и находится в файле **sh\_12F675\_kan1\_v1.zip**.

Предусмотрена установка транзисторов в разных исполнениях ТО-92 и SOT-23. МК устанавливается через переходную колодку DIN-8, что позволяет модифицировать и изменять программное обеспечение и возможные функции использования.

## Программное обеспечение

Для работы схемы замены используется программное обеспечение, совместимое с интерфейсом 1-Wire, которое подготовлено и представлено в [3]. Это программа термометрии **silos\_v3.exe**, которая написана на языке программирования FORTH

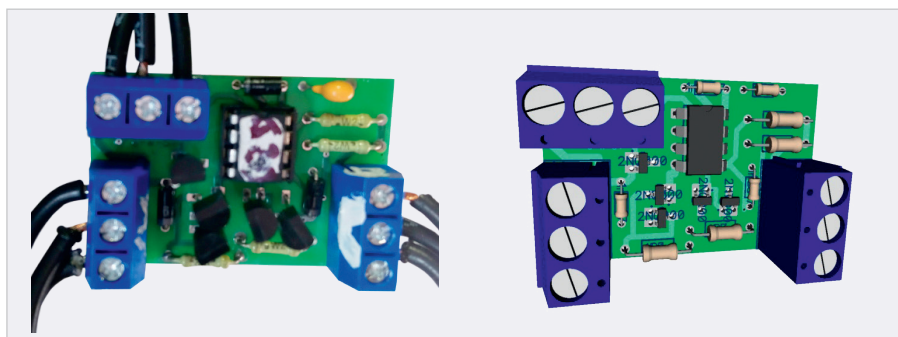


Рис. 2. Фото собранной платы замены DS2409 (слева) и её 3D-модель в DipTrace (справа)

[4]. Программа содержит все компоненты для программирования и модификации программы замены ds2409 [1] на 12F629 [2], а также печатные платы и схемы.

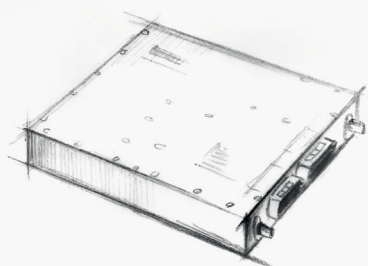
Программный код формируется в режиме «восстановление всех файлов – компиляция нового кода». Для получения кода и дальнейшей прошивки его через программатор требуется выполнить следующую последовательность действий с программой **silos\_v3.exe** [3]:

- скачать архив по адресу [3] и распаковать исполняемый файл **silos\_v3.exe** в отдельной папке;

- выполнить программу и в предлагаемом меню выбрать режим «восстановление всех исходных файлов» (нажать цифровую клавишу 3);
- программа выполнит восстановление всех исходных файлов и остановится;
- перейти в созданный каталог **test\_monsys\_exe** и выполнить командный файл **start\_new\_versii.cmd**;
- программа выполнит компиляцию нового кода, а также создаст каталог **hex\_12f629\_2409\_v1**, в котором созданы файлы для МК. Данные файлы имеют номера, которые привязаны к номерам устройств

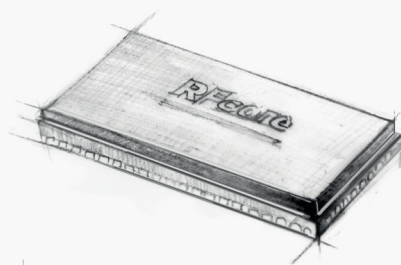


## НОВЫЕ МОЩНОСТИ — НОВЫЕ ВОЗМОЖНОСТИ



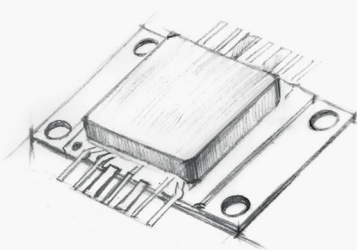
### СВЧ-усилители мощности

- Диапазон частот: от HF до Ku
- Выходная мощность: 2...1000 Вт
- Типовое усиление: 25...65 дБ
- Рабочее напряжение: 28, 40 В



### Многофункциональные CMOS MMIC

- Диапазон частот: S, C, X, Ku
- Выходная мощность: до 15 Вт
- Исполнение: QFN-корпус



### GaN и GaAs MMIC

- Диапазон частот: 2...18 ГГц
- Выходная мощность: до 12 Вт
- Типовое усиление: 10...23 дБ
- Исполнение: QFN-корпус/кристалл

**PROCHIP**  
POWERED BY PROSOFT

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА

(495) 232-2522 ■ INFO@PROCHIP.RU ■ WWW.PROCHIP.RU

Реклама

Таблица 1. Представление кода идентификации в прямом и обратном виде

1	2	3	4	5	6	7	8
D2	01	2F	62	90	04	0B	1F
1F	0B	04	90	62	2F	01	D2

идентификации протокола 1-Wire. Например, файл **hex\_12f629\_2409\_v1\_D2012F6290040B1F.hex** предназначен для схемы ветвителя с номером **D2012F6290040B1F**. Для работы программы термометрии этот номер записывается в файле инициализации к соответствующим датчикам DS1820, чтобы обращаться именно к данному ветвителю с установленными для него датчиками.

Полученные файлы нужно записать в МК любым программатором для данного типа МК. Если не требуется модернизация программы, например, изменение кодов для обращения к ветвителям, то сформированный каталог можно удалить. Если изменили коды обращения, то оставляете сформированный файл **silos\_v3.exe** и удаляете все предыдущие файлы. После компиляции сформированный файл **silos\_v3.exe** содержит все новые изменения для работы с новыми номерами идентификации ветвителя. В новом файле изменится контрольная сумма и дата создания, размер файла останется прежним.

При более «серьёзной» модификации программы изменится и размер исполняемого файла. Рассмотрим несколько поясняющих примеров функционирования программного обеспечения для схемы ветвителя на 12F629 на языке программирования FORTH. Текст программы находится в файле **silos\_v3\_2409\_v1.f** и компилируется через выполнение общей программы **silos\_v3.f** с помощью файла **100\_spf4.exe**. Для уточнения, где описание и где программа, фрагменты текста программ, «вырезанные» из основного текста, размещены в цветных квадратах.

### Пример 1. Изменение идентификационного кода

В стандарте интерфейса 1-Wire доступ к «ведомым» элементам осуществляется по их идентификационному коду, который состоит из 8 байт [1]. Первый байт определяет функцию или тип устройства, далее 6 байт каких-либо номеров, а последний байт – контрольная сумма первых 7 байт.

Таким образом, формируется стандартизация интерфейса 1-Wire и его

устройств. Например, первый байт 0x28 определяет датчики температур DS18B20, первый байт 0x1F определяет ветвителя DS2409 и так далее.

Требуется выполнять стандартизацию обращения к устройствам интерфейса 1-Wire, поскольку в этом случае программное обеспечение производителя будет работать нормально со схемой ветвителя на МК.

Следовательно, можно изменять со 2-го по 7-й байт и оставлять 1-й байт с кодом 0x1F. Последний байт формируется по первым 7 байтам.

Для формирования последовательности номеров задействованы 2-й и 3-й байт. Остальные 4, 5, 6, 7-й байты не меняются, и допустимо записывать любую информацию.

Рассмотрим распределение байт идентификационного номера:

**D2012F6290040B1F** – код идентификации, где:

- 1F – тип устройства 1-Wire, ветвитель DS2409, и он заменён на 12F629;
- 040B – номер, формируемый программой в количестве 10 штук, начиная с 0403 до 040C;
- 012F6290 – постоянные байты, где записан тип используемого микроконтроллера;
- D2 – контрольная сумма первых 7 байт.

Отсчёт нумерации байтов может выполняться в разных направлениях и зависит от представления в программе. На нижнем уровне передача начинается с байта кода информации об устройстве, а в программном обеспечении верхнего уровня отображается в обратном виде. Представление кодов идентификации с разным видом дано в табл. 1.

- В файле программы **silos\_v3\_2409\_v1.f** текстовым редактором находим строки, определяющие начальный номер идентификации:

```
CREATE ZAPROS_ACP_OUT_RGN1 0x3 C,
\мл. байт
CREATE ZAPROS_ACP_OUT_RGN2 0x4 C,
\ст. байт и меняем, например, старший байт на 0x1
```

Таким образом, будет формироваться последовательность 0103, 0104, ... 010C, всего 10 кодов.

Далее находим строку, определяющую количество файлов с идентификационными кодами:

```
CREATE KOЛVO_PIC_UT 0xA, \и изменяем на другое требуемое количество файлов, отличное от 10. Следует обращать внимание на различные способы записи чисел с разными форматами счисле-
```

ния. **0xA** – это число 10 в шестнадцатеричной системе счисления. Запись и отображение значения чисел контролируется через переменную **BASE**.

- В файле программы находим форт-слово **NAME\_PIC\_HEX\_RG1** и изменяем значение постоянных байтов.

Например, после **ks** на месте 12F629 пишем простой ряд чисел от 0...7:

```
:NAME_PIC_HEX_RG1 S> hex_12f629_2409_v1_ks012F629000001F.hex»; – был текст форт-слова,
```

```
: NAME_PIC_HEX_RG1 S> hex_12f629_2409_v1_ks0123456700001F.hex»; – стал текст форт-слова,
```

Необходимо отметить, что имя формируемого файла содержит уточнение функции работы. Это все байты до **ks**. При необходимости данные по названию тоже можно поменять. Например:

```
: NAME_PIC_HEX_RG1 S> hex_12f629_2409_v1_ks012F629000001F.hex»; – был текст форт-слова.
```

```
: NAME_PIC_HEX_RG1 S> Novosibirsk_SibGUTIS_ks012F629000001F.hex»; – стал текст форт-слова.
```

Чтобы не нарушать структуру имени файловой системы, размер текста не должен превышать 255 байт.

- Выполняем компиляцию программы через командный файл **start\_new\_versii.cmd** и получаем файлы для программирования МК. В процессе выполнения программа подсчитывает контрольную сумму **ks** и подставит её в нужное место. В байтах 2-м и 3-м формируются номера с увеличением на +1 для младшего байта. Все данные записываются в HEX-формате в шестнадцатеричной системе счисления.

Для понимания отличия значения числа и его отображения в различных системах счисления (сс) приведём пример отображения натуральных чисел в интервале 0...16 в сс от 16 до 2 на языке FORTH.

```
\ начало текста программы
: PKZ_DEMO BASE @ DUP
  DECIMAL CR
  S" Ncc=" TYPE . CR   BASE !
  2DUP DO I . LOOP ;
: FORMAT_CH_DEMO HEX 0x10 0
  2 0x10 DO I BASE ! PKZ_DEMO
  -1 +LOOP HEX 2DROP ;
  FORMAT_CH_DEMO
\ окончание текста программы.
```

Для работы программы нужно создать файл, например, **test.f** блокнотом, скопировать текст от комментария **\ начало** ... до комментария **\ окончание** ... и сохранить. Косая линия вле-

во – это начало комментария до конца строки, и также комментарии заключены в круглые скобки внутри форт-слов. Открыть файл **test.f** с помощью файла **100\_spf4.exe**, т.е. выполнить программу. Программа выполнится, и в окне консоли отобразится одинаковый ряд натуральных чисел в системе счисления 16...2. Компилятор работает в Windows/XP/7/8/10 и доступен без ограничений [4].

### Пример 2. Формирование низкоуровневых команд в конфигурации микроконтроллера

В форт-слове **START\_БЛОК** определяются используемые подпрограммы и настройки МК.

Например, две следующие строки определяют режим установки порта МК:

```
status 5 bcf (выбор банка 0 бит 5=0)
porta clrf (установка и сброс порта, выполняется инициализация режима фиксации сигналов)
```

Поясним определение первой строки: **status** – специальный регистр определен как **0x03 equ status**, согласно тех. описанию на МК;

5 – цифра, определяющая, какой бит установить в 0 для регистра **status**;

**bcf** – команда выполнения установки бита в указанном регистре.

Поясним определение **bcf**:

```
: bcf (status Ni – компилируем в область памяти команду по org уст. бита в 0)
```

```
|| status Ni || (локальные переменные)
```

```
-> Ni -> status (сохранили в локальных переменных)
```

```
0x1000 Ni 0x80 * + status + (Kkm – получили код команды)
```

```
WRITE_ORG (Kkm – запись и сдвиг на +1 и проверка на превышение);
```

В определяемом слове **bcf** слово **status** – это локальная переменная, а не специальный регистр. И значение локальной переменной приходит из входного стека, в нашем случае это 3 и 5.

Далее выполнется обработка для получения кода команды. Форт-слово **WRITE\_ORG** записывает этот код в область памяти.

Поясним определение **WRITE\_ORG**:

```
: WRITE_ORG (Kkm – запись и сдвиг на +1 и проверка на превышение)
```

```
0x100 /MOD SWAP 0x100 * + (перевернули мл. и ст. байты)
```

```
ORG 4+ @ ORG @ 2* + W! (записали словом в 2 байта)
```

```
ORG 1+! (переход к след. ячейке записи)
TEST_ORG_COMP (тестирование на превышение кол-ва компиляции ячеек);
```

В определяемом слове **WRITE\_ORG** арифметические операции подготовки кода и сдвиг на +1 байт счетчика кода в переменной **ORG**.

Поясним определение **TEST\_ORG\_COMP**:

```
: TEST_ORG_COMP (тестирование на превышение кол-ва компиляции ячеек)
```

```
ORG @ ORG 2 4* + @ = IF UST_ATR_LIM_CMD (установка атрибутов цветности знаков для тревоги)
```

```
CR ." превышен код компиляции
0x3FF FORTH-ASSEMBLER pic12f629 " (текст для тревоги)
```

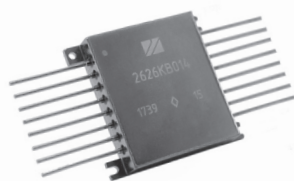
```
5 0 BEGIN BEEP2 0x100 PAUSE
exBEEP4 1+ 2DUP = UNTIL 2DROP (цикл 5 звуковых сигналов тревоги)
```

```
BYE (прекращение работы программы) THEN;
```

Используемая переменная **ORG** содержит значения текущей компиляции и

## НОВЫЕ ВОЗМОЖНОСТИ МИКРОСБОРОК ДЛЯ ПРЕДПРИЯТИЙ ОПК

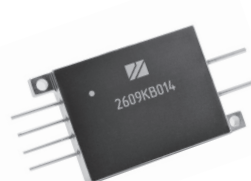
**ПРОТОН-ИМПУЛЬС**  
25 лет развития и роста



4×10 A / 100 VDC



65 A / 100 VDC



± 20 A / ± 400 V



3×25 A / 600 VAC

Для постоянного тока

Для двунаправленного и переменного тока

**Используются:**  
в аппаратуре спецназначения

**Назначение:**  
в качестве нормально разомкнутого ключа для коммутации цепей постоянного и переменного тока

**Заменяют:**  
электромагнитные реле малой и средней мощности, импортозамещение реле категории «Military»

РОССИЯ, г. ОРЕЛ, УЛ. ЛЕСКОВА, 19

(4862) 303-324 доб. 302

ENERGIA@PROTON-IMPULS.RU

WWW.PROTON-IMPULS.RU

возможное максимальное значение. Если они сравниваются, то выполнится IF-THEN – условие превышения, и на экране сформируется сообщение о превышении размера программы. Дополнительно прозвучит звуковой тревожный сигнал, и программа прекратит работу.

Все слова форт-ассемблерных команд МК доступны для модификации, что удобно при переходе к другим типам МК. Вторая строка данного примера определяется аналогично по тому же цепочному принципу поиска.

Описание цепочной структуры, т.е. определения форт-слов, показано в порядке последовательности работы.

В тексте программы определения расположены в обратном порядке, т.е. 2 строки примера завершают определение. Первым определяется переменная ORG, затем TEST\_ORG\_COMP и так далее. Завершает определение в нашем примере bcf.

На данном примере показано, что форт-слова допускают любую функцию или действие:

- исполняемое действие;
- компиляция в область памяти;
- указатель на переменную и значение.

Все форт-слова имеют цепочную связь, которая позволяет определить назначение, состав форт-слова, условия и возможности в работе.

### Пример 3. При отсутствии сигналов по шине 1-Wire более 2 секунд выполняется сброс ключей шины MAIN и AUX

В исходной микросхеме DS2409 такого режима нет, но поскольку существует вероятность сохранения подключения коммутируемой шины из-за помех при передаче, автор счёл необходимым добавить такую функцию.

Используются три общих регистра, которые определены так:

```
0x26 equ time_n0 \ счетчик 0г времени паузы,
0x27 equ time_n1 \ счетчик 1г времени паузы,
0x28 equ time_n2 \ счетчик 2г времени паузы; итого 3 байта 2r1r0r
```

Интервал отсутствия сигнала на шине 1-Wire задаётся в форт-слове UST\_TIME\_I2M\_PAUSE\_1W# и оформлен подпрограммой.

\ задаём значения подсчета в переменные регистры, начиная с младшего байта

```
0x40 movlw \ тут можно и FF или 0
```

```
time_n0 movwf \ счётчик 0г времени паузы
0x4B movlw \
time_n1 movwf \ счётчик 1г времени паузы
0x04 movlw \
time_n2 movwf \ счётчик 2г времени паузы 04 4b 40 = 2 сек; если сигнала нет, выдается уст. 0 main aux
```

В форт-слове ST\_RST\_1W находится участок, который отвечает за фиксацию начала сигнала RST. Там и расположен учёт отсутствия сигнала.

(ждём перепада до --\\_ и если очень долго ждём, то выдаём сигнал уст. aux=main=0, т.е. выкл.)

```
ORG @ m1 ! porta 0x3 btfss \ проверка на 0 gp3 выв.4 in-out-1w пропустить, если =1
```

```
m1 goto \ нет 1 далее, есть 0 возврат m1
```

```
1 UST_TIME_I2M_PAUSE_1W# (подпрограмма уст. значения для паузы)
```

```
ORG @ m2 ! 1 I2M_TIME_PAUSE_1W# (подсчёт по отниманию 1 и, если =0, то выкл. aux-main)
```

```
porta 0x3 btfsc \ проверка на 0 gp3 выв.4 in-out-1w; пропустить, если =0
```

```
m2 goto \ нет 0 далее, есть 1 возврат m2
```

Если программа ждёт сигнала 0, то она «циклится», и выполняется подпрограмма I2M\_TIME\_PAUSE\_1W#.

Поясним определение подпрограммы:

\ отнимаем 1 и смотрим, пока не будет всё 0, тогда выдаём сигнал выключения main aux

```
0x1 movlw \ значение 1 для вычитания в общий регистр
```

```
time_n0 0xF decfsz \ декремент вычесть 1 из f и пропустить, когда будет 0, т.е. к след. разряду
```

```
m10 goto \ уходим
```

```
time_n1 0xF decfsz \ декремент вычесть 1 из f и пропустить, когда будет 0, т.е. к след. разряду
```

```
m10 goto \ уходим
```

```
time_n2 0xF decfsz \ декремент вычесть 1 из f и пропустить, когда будет 0, т.е. выполним выкл.
```

```
m10 goto \ уходим
```

\ или выключаем main aux в случае всех 0 в time\_n0 n1 n2

```
porta 0x1 bcf (b'001m-x1aw' m/4/= a/1/= w/0/= - выбор bsf=1 bcf=0 нет aux)
```

```
porta 0x4 bcf (b'001m-x1aw' m/4/= a/1/= w/0/= - выбор bsf=1 bcf=0 нет main)
```

```
1 UST_TIME_I2M_PAUSE_1W# (подпрограмма уст. значения для паузы)
ORG @ m10 ! (выход)
```

Таким образом, при «зависании» или ожидании сигнала RST сброса от ведущего шины 1-Wire периодически выдаётся сигнал 0 на управляемые транзисторы для их закрытия.

Необходимо обратить внимание, что большая точность данного интервала не требуется, и можно значения для регистров time\_n0 time\_n1 не устанавливать. Время определяет регистр time\_n2, и примерное значение 1 равно задержке в 0,5 секунды. Более подробное изучение Форт-ассемблера выходит за рамки данной темы. Необходимо отметить, что в тексте программы имеются комментарии и пояснения по функционированию.

## Выводы

Предложенная схема и программа замены специальной микросхемы на универсальный микроконтроллер позволяет снизить монополизм производителей и фирм, поставляющих электронные компоненты.

Универсальный микроконтроллер создаёт возможность формирования новых функций к работающим системам измерений без их существенной модификации.

Использование языка программирования FORTH и подготовленного на этом языке FORTH-ASSEMBLER позволяет легко переходить на другие типы микроконтроллеров и также создавать программы с хорошей оптимизацией кодов для микроконтроллеров с малым объёмом памяти.

Язык программирования FORTH, кроме Windows, работает на всех основных типах операционных систем (Linux, Unix, Android и т.д.), существующих в настоящее время, что позволяет использовать данную программу на любых компьютерах.

## Литература

1. Описание DS2409 шины 1-Wire // URL: <http://www.gaw.ru/html.cgi/txt/ic/Maxim/1-Wire/start.htm>.
2. Описание 12F765 // URL: [http://www.microchip.ru/files/d-sheets-rus/PIC12F629\\_675.pdf](http://www.microchip.ru/files/d-sheets-rus/PIC12F629_675.pdf).
3. Каталог программы, платы: URL: [http://90.189.213.191:4422/doc\\_sh/toguchin\\_2020n/test/](http://90.189.213.191:4422/doc_sh/toguchin_2020n/test/) и копия в «облачном хранилище»: URL: [https://disk.yandex.ru/d/aKAO\\_DVD1LzrUQ](https://disk.yandex.ru/d/aKAO_DVD1LzrUQ).
4. Описание языка Форт spf4.exe, автор версии А. Черезов // URL: <http://www.forth.org.ru/>.

## НОВОСТИ МИРА

## КАК ИНТЕРНЕТ ВЕЩЕЙ ВМЕСТЕ С РАЗВИТИЕМ 5G И ПЕРИФЕРИЙНЫХ ВЫЧИСЛЕНИЙ СТИМУЛИРУЕТ ПРОМЫШЛЕННЫЕ ИННОВАЦИИ

Даже учитывая то, что количество подключённых устройств IoT в потребительском пространстве в настоящее время превышает количество подключённых устройств IoT в промышленности, инвестиции в промышленный IoT демонстрируют рост с точки зрения межотраслевых решений, а также устройств, предназначенных для удовлетворения потребностей конкретных секторов.

В исследовании консалтинговой фирмы Reply «Промышленный Интернет вещей: проверка реальности» рассмотрены две ключевые области, которые способствуют развитию IoT на промышленном рынке: интеллектуальное производство; интеллектуальный транспорт и логистика.

Соединяя оборудование и инструменты, промышленный Интернет вещей (IIoT) позволяет компаниям-производителям улучшить наглядность своего производства в режиме реального времени. Огромный объём данных, генерируемых устройствами IIoT, служит топливом для оптимизации производства, повышения качества доставки, внедрения профилактического обслуживания, автоматизации цепочки поставок и многого другого.

«Без промышленного Интернета вещей "Индустрия 4.0" не может существовать, – утверждает технический директор консалтинговой фирмы. – Данные – это топливо для всех "умных" сценариев использования в промышленном мире. Промышленный Интернет вещей – это ключевой элемент, который гарантирует инфраструктуру для сбора данных, их отправки в облако и управления пост-анализом обратной связи – всё это является частью целого круга преимуществ для бизнеса».

Несмотря на сложный экономический климат 2020 года, в обоих кластерах («Европа-5» – Германия, Италия, Франция, Бельгия и Нидерланды и «Большая пятёрка» – США, Китай, Индия, Бразилия и Великобритания) наблюдался небольшой, но заметный рост инвестиций в интеллектуальные предприятия, а также в область интеллектуального транспорта и логистики. К 2025 году ожидается дальнейший и гораздо более значительный рост. Ожидается, что к 2025 году объём кластера «Большой пятёрки» во главе с США превысит 86 млрд евро с серьёзными инвестициями в платформы, решениями для прогнозирования и удалённого мониторин-



га. Рынок интеллектуального транспорта и логистики должен превысить 15 млрд евро. В кластере «Европа-5», с другой стороны, ожидается, что рынок умных фабрик почти утроится во всех странах, достигнув в общей сложности более 23 миллиардов евро в пяти рассматриваемых странах во главе с Германией. Платформы настроены на экспоненциальный рост, и предприятия будут инвестировать в улучшение управления качеством и снижение затрат. Германия также останется лидером в области интеллектуального транспорта и логистики, но в других странах кластера по-прежнему будет наблюдаться значительный рост. Ожидается, что в 2025 году этот кластер достигнет общей стоимости 3,6 млрд евро.

Внедрение недорогих датчиков и сетей 5G, обусловленное крупными инвестициями телекоммуникационных компаний, ещё больше улучшит распространение промышленного Интернета вещей. Например, ожидается, что улучшенная связь между автономными транспортными средствами/роботами, искусственным интеллектом и механизмами в сочетании с повышенной вычислительной мощностью и очень низкой задержкой повысит не только эффективность предприятий, но и их безопасность. Кроме того, возможность создавать частные сети с высокой плотностью размещения позволит более широко использовать промышленный IoT, а также подключать значительное количество датчиков, оборудования, транспортных средств и роботов в дополнение к более широкому использованию дополненной и виртуальной реальности для поддержки «подключённых работников».

Постоянный рост количества подключённых устройств и их неоднородность требуют чёткого управления безопасностью настройки и политики обслуживания как устройств, так и сетей. Основываясь на своём опыте,

исследователи считают, что организациям необходимо внедрять микросегментированные среды (локальные и/или облачные), которые являются стабильными и готовыми реагировать как на традиционные, так и на новые опасные технологии и методы, тем самым снижая вероятность того, что новые виды атак будут успешными. Анализ архитектуры Интернета вещей, промышленных компонентов и всей инфраструктуры поможет компаниям заранее устранить существующие пробелы, уязвимости и угрозы. Но это гораздо больше, чем просто технологический вопрос: программы обучения сотрудников, наряду с изучением и постоянным тестированием всех используемых устройств, также будут иметь решающее значение.

Если в последние годы технологии промышленного Интернета вещей были приняты и использовались, прежде всего, для повышения эффективности заводов и логистических центров, то во время пандемии новые инвестиции были направлены в первую очередь на повышение безопасности работников. Однако ожидается, что долгосрочная тенденция напрямую затронет конечных потребителей. Успех так называемых «подключённых продуктов» на самом деле ускоряет инвестиции в решения, в которых сбор и обработка данных задействует не только производственное оборудование, но и использование готовой продукции. Модернизация процессов проектирования, производства и распространения продуктов, подключённых к Интернету вещей, позволяет создавать дополнительные услуги и облегчает возможность удалённого обновления и обслуживания бытовой техники, автомобилей, роботов, электроники и развлекательных устройств.

*vestnik-glonass.ru*