

возможное максимальное значение. Если они сравниваются, то выполняется 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 ведущей 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.
3. Каталог программы, платы: URL: http://90.189.213.191:4422/doc_sh/toguchin_2020n/test/ и копия в «облачном хранилище»: URL: https://disk.yandex.ru/d/aKAo_DVD1LzrUQ.
4. Описание языка Форт spf4.exe, автор версии А. Черезов // URL: <http://www.forth.org.ru/>.