Программная реализация I2C интерфейса (Master режим)

Статья основывается на технической документации DS00554c компании Microchip Technology Incorporated, USA.

тел. (095) 737-7545
www.microchip.ru

Автор: ООО "Микро-Чип"

Программная реализация I2C интерфейса (Master режим)

Источник: ООО “Микро-Чип” - Москва-2001
Статья основывается на технической документации DS00554c
компании Microchip Technology Incorporated, USA.

В статье описывается программная реализация I2C интерфейса в Master режиме для семейства микроконтроллеров PIC16CXXX, без использования аппаратной поддержки модуля SSP.

Краткое описание шины I2C можно прочитать в статье «Краткий обзор интерфейса I2C», за более детальной информацией обратитесь к документу №939839340011 “The I2C bus and how to use it” корпорации Philips.

На последовательной двухпроводной шине I2C, может быть несколько Master и Slave устройств соединенных друг с другом. Тактовый сигнал (линия SCL) формируется Master устройством, второй сигнал, данные (реверсивная линия SDA). Активный считается низкий уровень сигнала на линии, высокий (пассивный) уровень формируется подтягивающими резисторами.

Протокол шины I2C позволяет обнаруживать конфликты, синхронизировать тактовый сигнал, получать подтверждения о приеме данных.

В большинстве систем, микроконтроллер является Master, а периферийные внешние устройства Slave. Если это так, Вы можете использовать этот пример в своих разработках.

Режим конкуренции (Multi-Master) сложно реализовать программно, с выполнением всех требований спецификации I2C, без использования аппаратной поддержки (распознавание команд START, STOP и др.).

Для работы с шиной I2C используются макрокоманды (подпрограммы) высокого уровня описанные ниже. Как пример, испытательная программа для работы с двумя микросхемами EEPROM 24LC04 и 24LC01.

Работа с подпрограммами

Программное обеспечение работы с шиной I2C делиться на два уровня. Подпрограммы низкого уровня файл i2c_low.asm, и подпрограммы высокого уровня файл i2c_high.asm (см. тексты подпрограмм в файле 00554.zip).

Ниже приведены принятые сокращения:

S – условие START
P – условие STOP
SlvAR – Slave адрес (операция чтения)
SlvAW – Slave адрес (операция записи)
A – условие подтверждения (позитивное ACK)
N – условие подтверждения (негативное NACK)
D – байт данных, D[0] первый байт, D[1] второй байт

Формат команды

В подпрограммах высокого уровня поддерживаются основные структуры сообщений. Каждое Slave устройство поддерживает одну или несколько структур сообщений. Например, микросхема 24LC04 поддерживает следующее сообщение (чтобы записать байт в счетчик текущего адреса) S-SlvAW-A-D-A-P, что вызовет следующую последовательность действий:

- выполнить команду START
- передать Slave адрес
- ожидать подтверждения
- передать байт данных
- ожидать подтверждение
- выполнить команду STOP

Slave адрес

Поддерживается 7 и 10–разрядная адресация, согласно спецификации на шину I2C. Перед использованием подпрограмм должен быть загружен Slave адрес устройства, используя макрокоманды LOAD_ADDR_8 для 7-разрядно адресации и LOAD_ADDR_10 для 10-разрядной адресации Slave устройств. Эти макрокоманды не только загружаю адрес, но также включают нужный формат адресации (см. описание макрокоманд).

Удержание линии SCL

Тактовый сигнал на линии SCL всегда формирует Master устройства, но Slave может удерживать его в низком уровне, указывая Master в необходимости ожидания. Master устройство должно проверять это условие, и ожидать когда Slave отпустит линию SCL. Проверку удержания линии SCL можно включить/выключить установкой/сбросом бита _ENABLE_BUS_FREE_TIME. Если удержание линии будет длиться более 1 мс, то возникнет прерывание TOCKI указывающее на аварийную ситуацию.

Арбитраж

Функция арбитража необходима в системе с возможной конкуренцией устройств на шине I2C (режим Multi-Master), поэтому в данном примере не рассматривается.

Аппаратные средства

Для работы с шиной I2C используются два вывода тактовый сигнал SCL и сигнал данных SDA (в примере SCL – RB1, SDA – RB0).

Используемые порты ввода/вывода настраиваются при инициализации как входы (с третьем состоянием вывода), а в выходную защелку записывается нуль. Для формирования на линии сигнала низкого уровня соответствующий вывод настраивается как выход.

Регистры стояния и управления шины I2C

Регистр состояния Bus_Status

Бит 0: _Bus_Busy
1 = выполнена команда START
0 = условие STOP

Бит 1: _Abort
Если этот бит установлен, произошла фатальная ошибка, линия SCL удерживалась слишком долго. Сбрасывается программно.

Бит 2: _Txmt_Progress
1 = выполняется передача данных

Бит 3: _Rcv_Progress
1 = выполняется прием данных

Бит 4: _Txmt_Success
1 = передача данных завершена
0 = произошла ошибка

Бит 5: _Rcv_Success
1 = прием данных завершен
0 = произошла ошибка

Бит 6: _Fatal_error
1 = произошла фатальная ошибка (связь была прервана)

Бит 7: _ACK_error
1 = не получено подтверждение от Slave устройства

Управляющий регистр Bus_control

Биты этого регистра должны быть настроены программным обеспечением до выполнения действий на шине I2C. Некоторые подпрограммы высокого уровня изменяют биты этого регистра автоматически.

Бит 0: _10BitAddr
1 = 10-разрядная адресация
0 = 7-разрядная адресация

Бит 1: _Slave_RW
1 = операция чтения
0 = операция записи

Бит 2: _Last_Byte_Rcv
1 = последний принимаемый байт, подтверждение не выдается

Биты 3,4,5: не используются

Бит 6: _SlaveActive
Указывает на наличие Salve устройства на шине I2C, при использовании подпрограммы I2C_TEST_DEVICE

Бит 7: _TIME_OUT_
Указывает на удержание линии I2C более 1 мс

Подпрограммы низкого уровня

InitI2Cbus_Master
Первичная инициализация, настройка регистра статуса и управления, конфигурация линий SCL и SDA.

TxmtStartBit
Формирование на шине I2C команды START.

TxmtStopBit
Формирование на шине I2C команды STOP.

LOAD_ADDR_8
Загрузка 7-разрядного адреса Slave устройства.

LOAD_ADDR_10
Загрузка 10-разрядного адреса Slave устройства.

Txmt_Slave_Addr
Передача Slave адреса, в зависимости от последней выполненной команды LOAD_ADDR_8 или LOAD_ADDR_10, а также бита _Slave_RW.

SendData
Передача байта данных из регистра DataByte

GetData
Прием байта данных в регистр DataByte. Если бит _Last_Byte_Rcv установлен, подтверждение не формируется.

Подпрограммы высокого уровня (макросы)

Подпрограммы высокого уровня вызываю подпрограммы низкого уровня, описанные предварительно. В большинстве случаев необходимо только часть подпрограмм высокого уровня, что позволяет сэкономить память микроконтроллера.

I2C_TEST_DIVICE

Параметры:

Нет

Назначение:

Проверка наличия Slave устройства на шине I2C

Описание:

Перед использованием этой команды необходимо загрузить адрес Slave устройства по команде LOAD_ADDR_8 или LOAD_ADDR_10. Если устройство присутствует на шине будет установлен бит _SlaveActive в регистре Bus_Control

Сообщение:

S-SlvAW-A-P

Пример:

```
LOAD_ADDR_8         0xA0            ; адрес 24LC04
I2C_TEST_DEVICE
btfss               _SlaveActive
goto                SlaveNotPresent
```

I2C_WR

Параметры:

BYTES,_SourcePointer
_BYTES – количество передаваемых байт
_SourcePointer – адрес начала буфера в RAM

Назначение:

Передача блока данных

Описание:

Передает несколько байт (определяется параметром _BYTES_) в Slave устройство, начиная с адреса _SourcePointer. При возникновении ошибки передача блока прекращается.

Сообщение:

S-SlvAW-A-D[0]-A….A-D[N-1]-A-P

Пример:

```
btfsc       _Bus_Busy
goto        $-1
LOAD_ADDR_8 _Slave_1_Addr
I2C_WR      0x09,DataBegin
```

I2C_WR_SUB

Параметры:

BYTES,_SourcePointer_,_Sub_Adress_
_BYTES – количество передаваемых байт
SourcePointer – адрес начала буфера в RAM
_Sub_Adress_ - адрес в адресном пространстве Slave устройства

Назначение:

Передача блока данных с указанием адреса в Slave устройстве

Описание:

Передает несколько байт (определяется параметром _BYTES_) в Slave устройство, начиная с адреса _Sub_Adress_ из ОЗУ с адреса _SourcePointer_. Используется для Slave устройств в которых поддерживается инкремент внутреннего адреса. При возникновении ошибки передача блока прекращается.

Сообщение:

S-SlvAW-A-SubA-A-D[0]-A….A-D[N-1]-A-P

Пример:

```
LOAD_ADDR_8 _Slave_1_Addr
I2C_WR_SUB  0x08,DataBegin+1,0x30
```

I2C_WR_SWING

Параметры:

BYTES,_SourcePointer_,_Sub_Adress_
_BYTES – количество передаваемых байт
SourcePointer – адрес начала буфера в RAM
_Sub_Adress_ - адрес в адресном пространстве Slave устройства

Назначение:

Передача блока данных с указанием адреса в Slave устройстве

Описание:

Передает несколько байт (определяется параметром _BYTES_) в Slave устройство, начиная с адреса _Sub_Adress_ из ОЗУ с адреса _SourcePointer_. Используется для Slave устройств в которых не поддерживается инкремент внутреннего адреса. При возникновении ошибки передача блока прекращается.

Сообщение:

S-SlvAW-A-(SubA+0)-A-D[0]-A-P
S-SlvAW-A-(SubA+1)-A-D[1]-A-P
и т.д.

I2C_WR_BYTE_MEM

Параметры:

BYTES,_SourcePointer_,_Sub_Adress_
_BYTES – количество передаваемых байт
SourcePointer – адрес начала буфера в RAM
_Sub_Adress_ - адрес в адресном пространстве Slave устройства

Назначение:

Передача блока данных с указанием адреса в Slave устройстве

Описание:

Передает несколько байт (определяется параметром _BYTES_) в Slave устройство, начиная с адреса _Sub_Adress_ из ОЗУ с адреса _SourcePointer_. Используется для Slave устройств в которых не поддерживается инкремент внутреннего адреса, и требующее ожидания записи предыдущего байта. При возникновении ошибки передача блока прекращается.

Сообщение:

S-SlvAW-A-(SubA+0)-A-D[0]-A-P
Ожидание 1мс
S-SlvAW-A-(SubA+1)-A-D[1]-A-P
Ожидание 1мс
и т.д.

I2C_WR_BUF_MEM

Параметры:

BYTES,_SourcePointer_,_Sub_Adress_,_Device_BUF_SIZE_
_BYTES – количество передаваемых байт
SourcePointer – адрес начала буфера в RAM
_Sub_Adress_ - адрес в адресном пространстве Slave устройства
_Device_BUF_SIZE_ - размер буфера в Slave устройстве

Назначение:

Передача блока данных с указанием адреса в Slave устройстве и размера буфера

Описание:

Передает несколько байт (определяется параметром _BYTES_) в Slave устройство, начиная с адреса _Sub_Adress_ из ОЗУ с адреса _SourcePointer_. Используется для Slave устройств, в которых поддерживается инкремент внутреннего адреса, с разбиением всего блока передаваемых данных на части соответствующие буферу Slave устройства. При возникновении ошибки передача блока прекращается.

Сообщение:

Аналогичное макросу I2C_SUB_WR, только с разбиением на несколько посылок в соответствии с размером буфера Slave устройства.

I2C_READ

Параметры:

BYTES,_DestPointer_
_BYTES – количество принимаемых байт
DestPointer – адрес начала приемного буфера в RAM

Назначение:

Прием блока данных из Slave

Описание:

Принимает несколько байт (определяется параметром _BYTES_) из Slave устройства, и записывает в ОЗУ начиная с адреса _DestPointer_. При возникновении ошибки прием блока прекращается. При приеме последнего байта подтверждение не формируется.

Сообщение:

S-SlvAR-A-D[0]-A….A-D[N-1]-N-P

Пример:

```
LOAD_ADDR_10        _Slave_3_Addr
I2C_READ            8,DataBegin
btfss               _Rcv_Success
goto                ReceiveError
```

I2C_READ_SUB

Параметры:

BYTES,_DestPointer_,_Sub_Adress_
_BYTES – количество принимаемых байт
DestPointer – адрес начала приемного буфера в RAM
_Sub_Adress_ - адрес в адресном пространстве Slave устройства

Назначение:

Прием блока данных из Slave с указанного адреса

Описание:

Принимает несколько байт (определяется параметром _BYTES_) из Slave устройства начиная с адреса _Sub_Adress_, и записывает в ОЗУ начиная с адреса _DestPointer_. При возникновении ошибки прием блока прекращается. При приеме последнего байта подтверждение не формируется.

Сообщение:

S-SlvAW-A-SubAddr-A-S-SlvAR-A-D[0]-A….A-D[N-1]-N-P

Пример:

```
LOAD_ADDR_10        _Slave_3_Addr
I2C_READ_SUB        8,DataBegin, 0x60
btfss               _Rcv_Success
goto                ReceiveError
```

I2C_READ_BYTE или I2C_READ_STATUS

Параметры:

DestPointer
DestPointer – адрес начала приемного буфера в RAM

Назначение:

Прием байта из Slave устройства

Описание:

Принимает байт из Slave устройства, и записывает в ОЗУ по адресу _DestPointer_. При возникновении ошибки прием прекращается.

Сообщение:

S-SlvAR-A-D-N-P

I2C_WR_SUB_WR

Параметры:

Bytes1,_SrcPtr1_,_Sub_Addr_,_Bytes2_,ScrPtr2_
Bytes1 – количество передаваемых байт в первом блоке
SrcPtr1 – адрес начала буфера в RAM первого блока
SubAddr - адрес в адресном пространстве Slave устройства
Bytes2 – количество передаваемых байт во втором блоке
SrcPtr2 – адрес начала буфера в RAM второго блока

Назначение:

Передача двух блоков данных в одном сообщении с указанием адреса в Slave устройстве

Описание:

Передает два блока данных одним сообщением в Slave устройство, начиная с адреса _SubAddr_, без использования команды STOP. При возникновении ошибки передача прекращается.

Сообщение:

S-SlvAW-A-SubA-A-D1[0]-A….A-D1[N-1]-A-D2[0]-A….A-D2[M-1]-A-P

I2C_WR_SUB_RD

Параметры:

Count1,_SrcPtr_,_Sub_Addr_,_Count2_,_DestPtr_
Count1 – количество передаваемых байт
SrcPtr – адрес начала буфера в RAM блока передачи
SubAddr - адрес в адресном пространстве Slave устройства
Count2 – количество принимаемых байт
DestPtr – адрес начала приемного буфера в RAM

Назначение:

Передача и последующий прием блока данных в одном сообщении с указанием адреса в Slave устройстве

Описание:

Передает _Count1_ байт данных Slave устройство, начиная с адреса _SubAddr_, без использования команды STOP. Затем принимает _Count2_ байт. При возникновении ошибки обмен данными прекращается.

Сообщение:

S-SlvAW-A-SubA-A-D1[0]-A….A-D1[N-1]-A-S-SlvAR-A-D2[0]-A….A-D2[M-1]-N-P

I2C_WR_COM_WR

Параметры:

Count1,_SrcPtr1_,_Count2_,_SrcPtr2_
Count1 – количество передаваемых байт в первом блоке
SrcPtr1 – адрес начала буфера в RAM первого блока
Count2 – количество передаваемых байт во втором блоке
SrcPtr2 – адрес начала буфера в RAM второго блока

Назначение:

Передача двух блоков данных в одном сообщении в Slave устройство

Описание:

Передает два блока данных одним сообщением без указания адреса в Slave устройстве. При возникновении ошибки обмен данными прекращается. Может быть использован для управления LCD драйвером.

Сообщение:

S-SlvAW-A-D1[0]-A….A-D1[N-1]-A-D2[0]-A….A-D2[M-1]-A-P

Применение

I2C – простой в реализации двухпроводный интерфейс связи между микросхемами. Промышленностью выпускается большое количество микросхем управляемые по интерфейсу I2C (SEEPROM, RTC, расширители портов ввода/вывода, ЖКИ, АЦП и т.д.). Некоторые микроконтроллеры семейства PIC16CXXX не имеют аппаратной поддержки интерфейса I2C, но из-за высокой производительности (250 нс при 16МГц) он может быть реализован программно.

Исходные тексты подпрограмм низкого и высокого уровней – 00554.zip

Статья основывается на технической документации DS00554c
компании Microchip Technology Incorporated, USA.