Сопряжение с четырехразрядными светодиодными матричными (5*7) индикаторами типа HCMS-2xxx (КИПВ70А-4/5*7К, КИПВ71А-4/5*7К)

Поздравим себя с успехом — мы сформировали-таки массив данных, который нужно вывести в сдвиговый регистр индикатора, чтобы отобразить первую колонку этого символа. Осталась малость — вывести информацию в сдвиговый регистр. По сравнению с тем, что мы делали для формирования этого массива, вывод его в индикатор достаточно прост, что следует из приведенных ниже подпрограмм.

Фрагмент подпрограммы, осуществляющий вывод в индикатор информации из ячеек EKRAN - EKRAN+3:

```
;
    MOV   A,EKRAN+0
    ACALL  IND7BIT
;
    MOV   A,EKRAN+1
    ACALL  IND7BIT
;
    MOV   A,EKRAN+2
    ACALL  IND7BIT
;
    MOV   A,EKRAN+3
    ACALL  IND7BIT
;
;
    CLR   COLUMN0  ;ЗАСВЕТКА КОЛОНКИ
    MOV   A,COUNSK
    INC   A
    MOV   COUNSK,A
    LCALL  DEL1MS
    SETB   COLUMN0  ;ГАШЕНИЕ КОЛОНКИ
;
```

Подпрограмма, осуществляющая вывод в сдвиговый регистр 7 бит из выбранной ячейки, приведена ниже:

```
;
IND7BIT:
    RLC   A
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 7
    SETB   INDCLK
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 6
    SETB   INDCLK
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 5
    SETB   INDCLK
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 4
    SETB   INDCLK
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 3
    SETB   INDCLK
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 2
    SETB   INDCLK
;
    RLC   A
    MOV   INDLINE,C
    CLR   INDCLK   ;БИТ 1
    SETB   INDCLK
;
    RET
;
```

Непосредственно вывод осуществляется при помощи подпрограммы IND7BIT. Вначале мы помещаем в аккумулятор байт из ячейки EKRAN и вызываем эту подпрограмму, затем байт из ячейки EKRAN+1 с повторным вызовом подпрограммы, а затем и из ячеек EKRAN+2 и EKRAN+3. После этого командой CLR COLUMN0 открываем транзистор VT1, увеличиваем на 1 счетчик столбцов (теперь COUNSK будет содержать не 0, а 1), вызываем подпрограмму миллисекундной задержки (LCALL DEL1MS) и командой SETB COLUMN0 закрываем VT1. Все, мы отобразили первую колонку выводимой на индикатор цифры, увеличили COUNSK на 1 и теперь должны повторить все описанные действия для второй, третьей, четвертой и пятой колонок.

Ниже приведена подпрограмма вывода информации в HCMS-2xxx в завершенном виде.

```
;
ADOO   .EQU   30H
COUNSK .EQU   7BH
INDLINE .EQU   P1.5
INDCLK  .EQU   P1.6
EKRAN   .EQU   7CH
COLUMN1 .EQU   P1.0
COLUMN2 .EQU   P1.1
COLUMN3 .EQU   P1.2
COLUMN4 .EQU   P1.3
COLUMN5 .EQU   P1.4
;
;СОБСТВЕННО ПОДПРОГРАММА ВЫВОДА ИНФОРМАЦИИ
;НА HCMS-2XXX ИЗ ЯЧЕЕК С АДРЕСАМИ ОТ AD00
;ДО AD00+3
;
OTBHCMS:
;
    MOV   R7,#255
;
OTBHC1:
    MOV   R2,AD00+0
    MOV   R3,#0
    LCALL  VIVHCMS
;
    MOV   R2,AD00+1
    MOV   R3,#1
    LCALL  VIVHCMS
;
    MOV   R2,AD00+2
    MOV   R3,#2
    LCALL  VIVHCMS
;
    MOV   R2,AD00+3
    MOV   R3,#3
    LCALL  VIVHCMS
;
    DJNZ   R7,OTBHC1
    RET
;
;КОД СИМВОЛА В РЕГИСТРЕ R2, НОМЕР
;ЗНАКОМЕСТА ВЫВОДИМОГО СИМВОЛА В
;R3 — ОН МОЖЕТ БЫТЬ РАВЕН ЛИШЬ
;00, 01,02,03 И НИКАКОМУ ДРУГОМУ
;
VIVHCMS:
    MOV   A,#0
    MOV   COUNSK,A
    MOV   EKRAN,A
    MOV   EKRAN+1,A
    MOV   EKRAN+2,A
    MOV   EKRAN+3,A
;
    MOV   A,R3
```

```
    ANL   A,#00000011B
    MOV   R3,A
    MOV   A,#EKRAN
    ADD   A,R3
    MOV   R0,A   ;В R0 АДРЕС ТОЙ
;ИЗ EKRAN...EKRAN+3,
;КУДА ВЫВОД
    MOV   DPTR,#TABZNAK
    MOV   A,R2
    RL    A
    RL    A
    ADD   A,R2   ;A = 5*R2
    ADD   A,COUNSK ;A = 5*R2+COUNSK
    MOVC  A,@A+DPTR
    MOV   @R0,A  ;В EKRAN...EKRAN+3
;ВЫВЕЛИ БАЙТ, СООТВ.
;ПЕРВОЙ КОЛОНКЕ
;
    MOV   A,EKRAN+0
    ACALL IND7BIT
    MOV   A,EKRAN+1
    ACALL IND7BIT
    MOV   A,EKRAN+2
    ACALL IND7BIT
    MOV   A,EKRAN+3
    ACALL IND7BIT
;
    CLR   COLUMN1  ;ЗАСВЕТКА КОЛОНКИ
    MOV   A,COUNSK
    INC   A
    MOV   COUNSK,A
    LCALL DEL1MS
    SETB  COLUMN1  ;ГАШЕНИЕ КОЛОНКИ
;
;
    MOV   A,R3
    ANL   A,#00000011B
    MOV   R3,A
    MOV   A,#EKRAN
    ADD   A,R3
    MOV   R0,A   ;В R0 АДРЕС ТОЙ
;ИЗ EKRAN...EKRAN+3,
;КУДА ВЫВОД
    MOV   DPTR,#TABZNAK
    MOV   A,R2
    RL    A
    RL    A
    ADD   A,R2   ;A = 5*R2
    ADD   A,COUNSK ;A = 5*R2+COUNSK
    MOVC  A,@A+DPTR
    MOV   @R0,A  ;В EKRAN...EKRAN+3
;ВЫВЕЛИ БАЙТ, СООТВ.
;ВТОРОЙ КОЛОНКЕ
;
    MOV   A,EKRAN+0
    ACALL IND7BIT
    MOV   A,EKRAN+1
    ACALL IND7BIT
    MOV   A,EKRAN+2
    ACALL IND7BIT
    MOV   A,EKRAN+3
    ACALL IND7BIT
;
    CLR   COLUMN2  ;ЗАСВЕТКА КОЛОНКИ
    MOV   A,COUNSK
    INC   A
    MOV   COUNSK,A
    LCALL DEL1MS
    SETB  COLUMN2  ;ГАШЕНИЕ КОЛОНКИ
;
;
    MOV   A,R3
    ANL   A,#00000011B
    MOV   R3,A
    MOV   A,#EKRAN
    ADD   A,R3
    MOV   R0,A   ;В R0 АДРЕС ТОЙ
;ИЗ EKRAN...EKRAN+3,
;КУДА ВЫВОД
    MOV   DPTR,#TABZNAK
    MOV   A,R2
    RL    A
    RL    A
    ADD   A,R2   ;A = 5*R2
    ADD   A,COUNSK ;A = 5*R2+COUNSK
    MOVC  A,@A+DPTR
    MOV   @R0,A  ;В EKRAN...EKRAN+3
;ВЫВЕЛИ БАЙТ, СООТВ.
;ТРЕТЬЕЙ КОЛОНКЕ
;
    MOV   A,EKRAN+0
    ACALL IND7BIT
    MOV   A,EKRAN+1
    ACALL IND7BIT
    MOV   A,EKRAN+2
    ACALL IND7BIT
    MOV   A,EKRAN+3
    ACALL IND7BIT
;
;
    CLR   COLUMN3  ;ЗАСВЕТКА КОЛОНКИ
    MOV   A,COUNSK
    INC   A
    MOV   COUNSK,A
    LCALL DEL1MS
    SETB  COLUMN3  ;ГАШЕНИЕ КОЛОНКИ
;
;
    MOV   A,R3
    ANL   A,#00000011B
    MOV   R3,A
    MOV   A,#EKRAN
    ADD   A,R3
    MOV   R0,A   ;В R0 АДРЕС ТОЙ
;ИЗ EKRAN...EKRAN+3,
;КУДА ВЫВОД
    MOV   DPTR,#TABZNAK
    MOV   A,R2
    RL    A
    RL    A
    ADD   A,R2   ;A = 5*R2
    ADD   A,COUNSK ;A = 5*R2+COUNSK
    MOVC  A,@A+DPTR
    MOV   @R0,A  ;В EKRAN...EKRAN+3
;ВЫВЕЛИ БАЙТ, СООТВ.
;ЧЕТВЕРТОЙ КОЛОНКЕ
;
    MOV   A,EKRAN+0
    ACALL IND7BIT
    MOV   A,EKRAN+1
    ACALL IND7BIT
    MOV   A,EKRAN+2
    ACALL IND7BIT
    MOV   A,EKRAN+3
    ACALL IND7BIT
;
    CLR   COLUMN4  ;ЗАСВЕТКА КОЛОНКИ
    MOV   A,COUNSK
    INC   A
    MOV   COUNSK,A
    LCALL DEL1MS
    SETB  COLUMN4  ;ГАШЕНИЕ КОЛОНКИ
;
;
    MOV   A,R3
    ANL   A,#00000011B
    MOV   R3,A
    MOV   A,#EKRAN
    ADD   A,R3
    MOV   R0,A   ;В R0 АДРЕС ТОЙ
;ИЗ EKRAN...EKRAN+3,
;КУДА ВЫВОД
    MOV   DPTR,#TABZNAK
    MOV   A,R2
    RL    A
    RL    A
    ADD   A,R2   ;A = 5*R2
    ADD   A,COUNSK ;A = 5*R2+COUNSK
    MOVC  A,@A+DPTR
    MOV   @R0,A  ;В EKRAN...EKRAN+3
;ВЫВЕЛИ БАЙТ, СООТВ.
```

```
;ПЯТОЙ КОЛОНКЕ
;
     MOV    A,EKRAN+0
     ACALL  IND7BIT
     MOV    A,EKRAN+1
     ACALL  IND7BIT
     MOV    A,EKRAN+2
     ACALL  IND7BIT
     MOV    A,EKRAN+3
     ACALL  IND7BIT
;
     CLR    COLUMN5  ;ЗАСВЕТКА КОЛОНКИ
     MOV    A,COUNSK
     INC    A
     MOV    COUNSK,A
     LCALL  DEL1MS
     SETB   COLUMN5  ;ГАШЕНИЕ КОЛОНКИ
;
;
     RET
;
;ПОДПРОГРАММА ВЫВОДА НА ДИСПЛЕЙ
;ОДНОГО СТОЛБЦА
;
IND7BIT:
     RLC    A
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 7
     SETB   INDCLK
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 6
     SETB   INDCLK
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 5
     SETB   INDCLK
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 4
     SETB   INDCLK
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 3
     SETB   INDCLK
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 2
     SETB   INDCLK
     RLC    A
     MOV    INDLINE,C
     CLR    INDCLK    ;БИТ 1
     SETB   INDCLK
     RET
;
;ЗНАКОГЕНЕРАТОР НА 32 СИМВОЛА
;
TABZNAK:
;
;ЦИФРА 0 [00H]
.DB     10011110B,10100001B,10100001B,10011110B,10000000B
;ЦИФРА 1 [01H]
.DB     10000000B,10000000B,10000010B,10111111B,10000000B
;ЦИФРА 2 [02H]
.DB     10100010B,10110001B,10101001B,10100110B,10000000B
;ЦИФРА 3 [03H]
.DB     10010001B,10100101B,10100111B,10011001B,10000000B
;ЦИФРА 4 [04H]
.DB     10001111B,10001000B,10001000B,10111111B,10000000B
;ЦИФРА 5 [05H]
.DB     10010111B,10100101B,10100101B,10011001B,10000000B
;ЦИФРА 6 [06H]
.DB     10011110B,10100101B,10100101B,10011001B,10000000B
;ЦИФРА 7 [07H]
.DB     10100001B,10010001B,10001001B,10000111B,10000000B
;ЦИФРА 8 [08H]
.DB     10011010B,10100101B,10100101B,10011010B,10000000B
;ЦИФРА 9 [09H]
.DB     10100110B,10101001B,10101001B,10011110B,10000000B
;ЦИФРА A [0AH]
.DB     10011110B,10100001B,10100001B,10011110B,10000000B
;ЦИФРА B [0BH]
.DB     10011110B,10100001B,10100001B,10011110B,10000000B
;ЦИФРА C [0CH]
.DB     10011110B,10100001B,10100001B,10011110B,10000000B
;ЦИФРА D [0DH]
.DB     10011110B,10100001B,10100001B,10011110B,10000000B
;ЦИФРА E [0EH]
.DB     10011110B,10100001B,10100001B,10011110B,10000000B
;ЦИФРА F [0FH]
.DB     10000000B,10000000B,10000000B,10000000B,10000000B
;
;ЦИФРА 0,[10H]
.DB     10011110B,10100001B,10100001B,10011110B,11100000B
;ЦИФРА 1,[11H]
.DB     10000000B,10000000B,10000010B,10111111B,11100000B
;ЦИФРА 2,[12H]
.DB     10100010B,10110001B,10101001B,10100110B,11100000B
;ЦИФРА 3,[13H]
.DB     10010001B,10100101B,10100111B,10011001B,11100000B
;ЦИФРА 4,[14H]
.DB     10001111B,10001000B,10001000B,10111111B,11100000B
;ЦИФРА 5,[15H]
.DB     10010111B,10100101B,10100101B,10011001B,11100000B
;ЦИФРА 6,[16H]
.DB     10011110B,10100101B,10100101B,10011001B,11100000B
;ЦИФРА 7,[17H]
.DB     10100001B,10010001B,10001001B,10000111B,11100000B
;ЦИФРА 8,[18H]
.DB     10011010B,10100101B,10100101B,10011010B,11100000B
;ЦИФРА 9,[19H]
.DB     10100110B,10101001B,10101001B,10011110B,11100000B
;ЦИФРА A,[1AH]
.DB     10011110B,10100001B,10100001B,10011110B,11100000B
;ЦИФРА B,[1BH]
.DB     10011110B,10100001B,10100001B,10011110B,11100000B
;ЦИФРА C,[1CH]
.DB     10011110B,10100001B,10100001B,10011110B,11100000B
;ЦИФРА D,[1DH]
.DB     10011110B,10100001B,10100001B,10011110B,11100000B
;ЦИФРА E,[1EH]
.DB     10011110B,10100001B,10100001B,10011110B,11100000B
;ЦИФРА F,[1FH]
.DB     10000000B,10000000B,10000000B,10000000B,10000000B
;
;
DEL1MS: MOV    R7,#07FH
LREX:   MOV    R6,#001H
LRIN:   DJNZ   R6,LRIN
     DJNZ   R7,LREX
     RET
;
```

В итоге у нас должна получиться такая подпрограмма вывода информации в HCMS-2xxx (вместе с необходимыми для ее функционирования дополнительными подпрограммками), как та, что приведена выше. Она вполне работоспособна, и на этом можно было бы остановиться. Но…

Посмотрите еще раз внимательно на подпрограмму VIVHCMS. Обратите внимание на то, что в ней есть пять больших повторяющихся фрагментов, начинающихся с команды MOV A, R3 и заканчивающихся командой ACALL IND7BIT, идущей после MOV A, EKRAN+3. Очевидно, если мы выделим этот повторяющийся фрагмент в подпрограмму (назовем ее, к примеру, VIVHCM1), то главное наше детище, подпрограмма VIVHCMS станет гораздо короче и изящнее. Давайте попробуем это сделать.

```
;
VIVHCMS:
     MOV    A,#0
     MOV    COUNSK,A
     MOV    EKRAN,A
     MOV    EKRAN+1,A
     MOV    EKRAN+2,A
     MOV    EKRAN+3,A
;
;
```

```
    LCALL  VIVHCM1
;
    CLR    COLUMN1  ;ЗАСВЕТКА КОЛОНКИ
    MOV    A,COUNSK
    INC    A
    MOV    COUNSK,A
    LCALL  DEL1MS
    SETB   COLUMN1  ;ГАШЕНИЕ КОЛОНКИ
;
;
    LCALL  VIVHCM1
;
    CLR    COLUMN2  ;ЗАСВЕТКА КОЛОНКИ
    MOV    A,COUNSK
    INC    A
    MOV    COUNSK,A
    LCALL  DEL1MS
    SETB   COLUMN2  ;ГАШЕНИЕ КОЛОНКИ
;
;
    LCALL  VIVHCM1
;
    CLR    COLUMN3  ;ЗАСВЕТКА КОЛОНКИ
    MOV    A,COUNSK
    INC    A
    MOV    COUNSK,A
    LCALL  DEL1MS
    SETB   COLUMN3  ;ГАШЕНИЕ КОЛОНКИ
;
;
    LCALL  VIVHCM1
;
    CLR    COLUMN4  ;ЗАСВЕТКА КОЛОНКИ
    MOV    A,COUNSK
    INC    A
    MOV    COUNSK,A
    LCALL  DEL1MS
    SETB   COLUMN4  ;ГАШЕНИЕ КОЛОНКИ
;
;
    LCALL  VIVHCM1
;
    CLR    COLUMN5  ;ЗАСВЕТКА КОЛОНКИ
    MOV    A,COUNSK
    INC    A
    MOV    COUNSK,A
    LCALL  DEL1MS
    SETB   COLUMN5  ;ГАШЕНИЕ КОЛОНКИ
;
    RET
;
VIVHCM1:
    ANL    A,#00000011B
    MOV    R3,A
    MOV    A,#EKRAN
    ADD    A,R3
    MOV    R0,A    ;В R0 АДРЕС ТОЙ
;ИЗ EKRAN...EKRAN+3,
;КУДА ВЫВОД
    MOV    DPTR,#TABZNAK
    MOV    A,R2
    RL     A
    RL     A
    ADD    A,R2    ;A = 5*R2
    ADD    A,COUNSK ;A = 5*R2+COUNSK
    MOVC   A,@A+DPTR
    MOV    @R0,A   ;В EKRAN...EKRAN+3
;ВЫВЕЛИ БАЙТ, СООТВ.
;КОЛОНКЕ
;
    MOV    A,EKRAN+0
    ACALL  IND7BIT
    MOV    A,EKRAN+1
    ACALL  IND7BIT
    MOV    A,EKRAN+2
    ACALL  IND7BIT
    MOV    A,EKRAN+3
    ACALL  IND7BIT
;
    RET
;
```

Полученный сокращенный вариант подпрограммы VIVHCMS приведен выше. Как видите, он гораздо короче, проще и нагляднее. Надо сказать, что обычно программы даже у опытных программистов никогда сразу не получаются оптимальными — поначалу не до оптимизации, нужно хотя бы заставить их просто работать без ошибок. Но очень часто, к сожалению, после того, как все заработает нужным образом, нам лень “навести блеск” — на это не остается ни сил, ни времени. Однако все же к этому нужно стремиться — по опыту знаю, что впоследствии такую программу гораздо проще дорабатывать. А заниматься доработками приходится гораздо чаще, чем это можно было бы предположить. Так что не стремитесь наступать на грабли обязательно самостоятельно, учитесь на чужих ошибках.

Краткие выводы

Мы познакомились с тем, как сопрягать микроконтроллер с различными типами знакосинтезирующих индикаторов. Как уже упоминалось выше, последние могут быть жидкокристаллическими или светодиодными (остальные, например, люминисцентные или электроннолучевые, практически вышли из употребления). По способу формирования символов индикаторы могут быть семисегментными или матричными. Далее, как мы имели возможность убедиться, многие из них, в особенности жидкокристаллические, снабжены самостоятельными микроконтроллерами, заметно упрощающими работу с ними. Именно такими были оба рассмотренных в настоящей главе жидкокристаллических индикатора. В обоих случаях нам пришлось лишь организовать передачу информации от ос-

новного МК в контроллер индикатора, а все, что связано с отображением переданных символов, осуществлялось уже без нашего вмешательства. Светодиодные индикаторы, которые мы рассмотрели, не имели встроенных контроллеров. При этом для работы с АЛС318 нам пришлось использовать в цепях сопряжения несколько дополнительных микросхем, а для HCMS-2xxx потребовались пять транзисторов. Программы, написанные для работы со светодиодными индикаторами, оказались заметно сложнее, чем аналогичные для ЖК-индикаторов.

Теперь уже не только мне, но и вам должно быть очевидно, что каких-либо стандартных правил сопряжения МК с индикаторами не существует, и в каждом конкретном случае оно выполняется по-своему. Поэтому мы и рассмотрели столь отличающиеся друг от друга варианты индикаторов, каждый со своей схемой включения, алгоритмом и программой сопряжения. После знакомства с предложенным материалом, я надеюсь, вы не только сможете любой из этих индикаторов использовать в дальнейшем в своих устройствах, но и опираясь на эти наработки самостоятельно сопряжете ваш МК с любым другим устройством индикации.

Кроме того, мы попытались попрактиковаться в разбиении поставленной задачи на ряд более простых, и в написании подпрограмм для этих простых задач. Пока у вас еще просто очень мало подобных навыков, и описанные в этой главе примеры, равно как и те, которые вы встретите в последующих главах, направлены в первую очередь на то, чтобы эти навыки у вас сформировать. Я попытался на практике показать вам, как разбивать задачу на части, как решать каждую из этих малых задач, причем даже не сразу, а, как говорится, в два-три приема. Я специально показал вам, что получается при первой попытке решения задачи, и как оно дальше оптимизируется. Те, кто имеет большой опыт, проводит эти предварительные стадии разработки в уме. Но у нас с вами такого опыта пока еще нет, поэтому мы решали нашу задачу постепенно, получив в конце концов как некоторые новые навыки, так и правильно работающую подпрограмму.

Еще раз обращаю ваше внимание на то, что любые действия, которые вам придется делать в вашей программе более, чем один или два раза, полезно оформить в виде подпрограммы со сразу понятным для вас именем. Именно так мы и поступали в рассмотренных примерах. При организации вывода информации на экран HT1610 мы создали подпрограмму SIMBOL1, после чего заносили в аккумулятор числа из AD00+7, AD00+6 и т. д., вызывая эту подпрограмму SIMBOL1 после каждого занесения. В случае с АЛС318 мы создали подпрограмму DISPLEY, при вызове которой в R0 мы помещали число от 0 до 7, и она далее извлекала цифру из ячейки памяти МК с адресом AD00+n (n=0-7), и выводила его в порт P1 для отображения в n-й разряд АЛС318. Мы написали подпрограмму VIVHCMS для вывода на HCMS-2xxx символа, помещенного в регистр R2 в разряд индикатора, номер которого занесен в регистр R3. С ее помощью выводить символы в HCMS-2xxx стало легко и просто.

Как видите, по крайней мере в тех случаях, когда вам нужно вывести на тот или иной дисплей цифровую информацию (а это всегда не менее 3-4 цифр), очень полезно выделить в самостоятельную подпрограмму все действия по пересылке символа на дисплей, при этом сам пересылаемый символ (а если надо, то и его место отображения на индикаторе) должны находится в каком-нибудь регистре (регистрах) вашего МК. Если вы привыкнете это делать, то написанные вами программы будут проще, стройнее и будут содержать меньше ошибок, чем если вы будете писать их без подобной привычки.

Александр Фрунзе
alex.fru@mtu-net.ru