

# Способ кодирования данных при передаче по телекоммуникационным линиям связи

Существует два способа кодирования данных для передачи по телекоммуникационным последовательным линиям связи CAN, LIN, I<sup>2</sup>C, UART, USB, SPI, I<sup>2</sup>S: либо каждый бит данных сопровождают синхросигналом, передаваемым по второй, дополнительной линии, либо данным придают свойство префиксности с помощью признака для безошибочного разделения соседних битов друг от друга на стороне приемника. Такие коды еще называют самосинхронизирующимися.

В статье предложен способ кодирования данных добавлением признака префиксности. Передача ведется по двум равносильным информационным линиям, причем по первой линии передаются данные, по второй — не синхросигналы, а префиксная последовательность, формируемая с учетом данных по определенному правилу.

Предложенный способ позволяет либо снизить технические требования к качеству обеих линий, либо повысить скорость и вероятность достоверной передачи в существующих телекоммуникационных линиях связи.

Виктор САФРОНОВ,  
к. т. н.

## Введение

Первая проблема для приемника, принимающего данные, передаваемые по телекоммуникационной линии связи (в последовательном интерфейсе), заключается в умении безошибочно разделять между собой принимаемые соседние биты. При этом известные способы кодирования различаются, по существу, способами решения коллизии — разделения соседних битов с одинаковыми значениями: «нулями» или «единицами», следующими подряд друг за другом. Коды, способные решить эту коллизию, принято называть самосинхронизирующимися (СК).

Вторая проблема для приемника — наличие или отсутствие постоянной (средней) составляющей уровня сигнала в принятой посылке. Если ее нет, посылку можно считывать и запоминать, например, на магнитном или оптическом носителе, принимать и передавать в телекоммуникационную линию связи через гальваническую трансформаторную развязку и т. д.

Для решения обеих проблем используют либо предварительное кодирование данных, добавляющее свойство префиксности, то есть критерий для безошибочного разделения соседних битов друг от друга на стороне приемника, либо простое сопровождение каждого передаваемого бита сигналом синхронизации по дополнительной телекоммуникационной линии.

Существует еще аппаратный способ организации передачи данных только по одной линии, без линии для синхросигналов, например в компьютерных UART-интерфейсах COM с протоколами RS-232, RS-485, RS-422 и т. д. Вместо дополнительной квитирующей линии в таком случае требуется предварительная жесткая договоренность общающихся сторон о выборе равных частот обоих синхрогенераторов. Но этого мало: нужна еще и кварцевая стабилизация частот синхрогенераторов. Кроме того, в посылку встраивают старт-стопные биты для фазовой автоподстройки синхрогенераторов обеих сторон.

Только так можно добиться надежного приема. Для оперативной дуплексной передачи разнородных массивов информации на большие расстояния в условиях помех подобный способ малоприменим, учитывая, что передающая и приемная стороны должны предварительно договориться о равных частотах синхрогенераторов и о количестве старт-стопных битов в посылках.

## Известные способы кодирования данных

Известно [1] много кодов для передачи данных по последовательному каналу связи, некоторые из них показаны на рис. 1 (кроме 9 и 10 графиков).

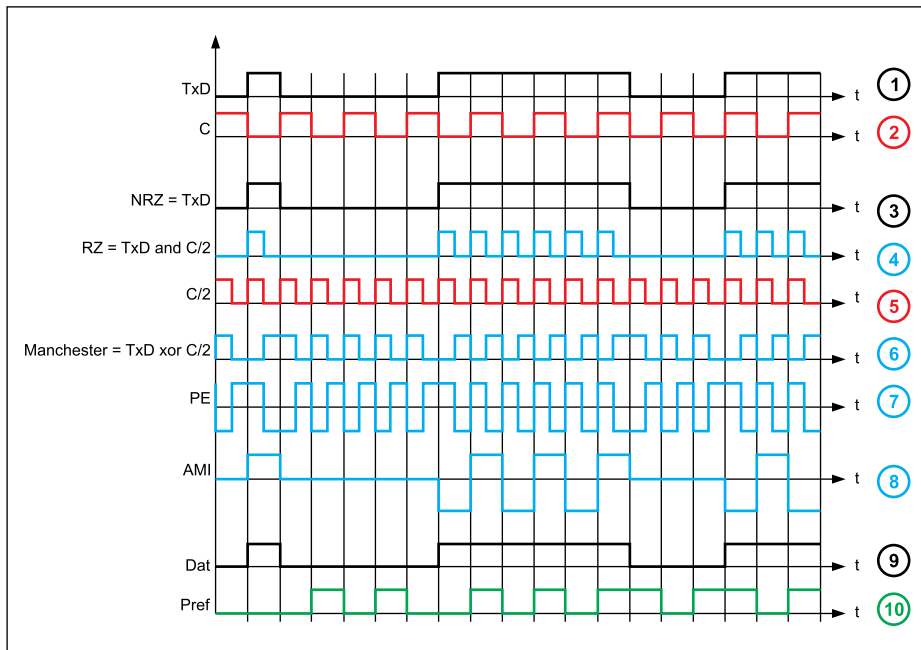
Из всех приведенных на рис. 1 известных кодов только Manchester и PE обладают свойством префиксности (самосинхронизации). Остальные коды для разделения битов требуют передачи импульсов синхронизации (по второй линии).

Предлагаемый код в виде пары линий (рис. 1, 9 и 10 графики) тоже имеет свойство префиксности.

## Скремблирование

Это еще один способ кодирования данных, представляющий обратимое преобразование структуры цифрового потока без изменения скорости передачи с целью получения свойств случайной последовательности. Скремблер (перемешиватель) реализует логическую операцию mod2 (сложение по модулю два) передаваемой последовательности с дополнительно формируемой псевдослучайной последовательностью.

Для выделения информации из принятой последовательности на стороне приемника в дескремблере производится еще раз логическая операция mod2 (сложение по модулю два) над принятыми символами.



**Рис. 1.** Известные способы кодирования данных (кроме двух нижних графиков):  
 TxD — передаваемые данные;  
 C — сигнал синхронизации (вариант 1);  
 NRZ (Non Return to Zero) — способ кодирования «без возвращения к нулю»;  
 RZ (Return to Zero) — способ кодирования «с возвращением к нулю»;  
 C/2 — сигнал синхронизации удвоенной частоты (вариант 2);  
 PE (Phase Encode) — способ фазового кодирования;  
 AMI (Alternate Mark Inversion) — способ биполярного кодирования с поочередной инверсией уровней;  
 Dat, Pref — предлагаемый способ кодирования двумя сигналами

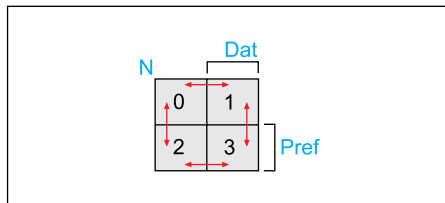
Различают два основных типа скремблеров-дескремблеров: самосинхронизирующиеся и с начальной установкой (аддитивные). Самосинхронизирующемуся скремблированию свойственно размножение ошибок после допущенной первой ошибки. При аддитивном скремблировании этих ошибок нет, однако требуется предварительная идентичная установка состояний регистров скремблера и дескремблера.

**Описание предлагаемого способа кодирования данных**

- Целями остальной части статьи являются:
- разработка способа кодирования и декодирования;
- получение префиксной последовательности, придающей пересылаемым данным свойство самосинхронизации;
- синтез синхронного автомата для осуществления процесса кодирования.

Предлагаемый способ кодирования опишем по таблице кодирования состояний (рис. 2).

В таблице красными стрелками показаны направления переходов состояний при изменении кодирующих переменных. Соседние переходы закодированы кодом Грея. Это позволяет, контролируя внутренние состояния логической операцией mod2, всегда судить о правильности кодирования на стороне передачи и о работоспособности на стороне



**Рис. 2.** Таблица кодирования состояний:  
 N — номера состояний (0, 1, 2, 3);  
 кодирующие переменные:  
 Dat — текущее значение принятого бита;  
 Pref — значение префиксного сигнала

приема. Кроме того, это же позволяет приемнику однозначно отделять бит от бита и декодировать информацию.

Закодированные данные, как было сказано, физически представляют собой две последовательности: собственно поток (посылка) данных и префиксный поток (посылка), сформированный с учетом данных по нижеприведенному правилу.

По этой таблице опишем процесс кодирования, то есть формирования сигналов Pref префиксной посылки, по отдельной линии сопровождающей пересылаемые данные.

Пусть, например, синхронный автомат находится в состоянии «0» и на него поступает следующий Dat-бит с «1» значением. Тогда автомат переходит в состояние «1». Если и в следующий момент времени опять придет «1» бит, то автомат перейдет в состояние «3»,

иначе — в состоянии «0». Сколь угодно длительная посылка, сплошь состоящая из битов с «1» значениями, будет заставлять автомат переключаться между состояниями «1» и «3». Именно эти два состояния обеспечивают свойство префиксности, необходимое для разделения в приемнике друг от друга битов посылки с одинаковыми значениями.

Точно такая же ситуация возникнет в состоянии «0» автомата и при подаче посылки со сплошными «0» значениями битов. Только теперь переключения автомата будут происходить между состояниями «0» и «2».

Переходы между состояниями «3» и «2» гарантируют сохранение свойства префиксности при смене сплошного потока «1» значений на сплошной поток «0» значений текущих битов. Остальные случаи кодирования битов последовательности тривиальны и понятны из таблицы кодирования состояний.

Таблице кодирования состояний (рис. 2) соответствует синхронная таблица переходов (рис. 3).

		TxD		
NN+		02	01	
0		20	23	} Pref
2		32	31	
3		10	13	} Dat
1				

**Рис. 3.** Синхронная таблица переходов:  
 NN+ — переход из указанного предыдущего состояния в указанное последующее состояние (парные цифры);  
 TxD — значение текущего бита на линии связи;  
 Pref — кодирующая переменная, значение формируемого префиксного бита;  
 Dat — кодирующая переменная, значение бита, принятого ранее

В синхронной таблице переходов NN+ показаны пары: текущее и следующее состояние автомата, соответствующие наборам кодирующих переменных Dat и Pref и воздействию текущему значению бита TxD.

**Построение логической схемы синхронного автомата для кодирования данных**

По таблице (рис. 3) выпишем (тоже парами) соответствующие текущие и последующие значения кодирующих переменных

Dat	Dat+	TxD		Pref	Pref+	TxD	
0	00	01	00	0	01	00	00
2	00	01	00	2	10	11	Pref
3	10	11	10	3	11	10	} Dat
1	10	11	10	1	00	01	

**Рис. 4.** Таблицы изменения значений кодирующих переменных Dat и Pref

QQ+	J	K	D
00	0	x	0
01	1	x	1
10	x	1	0
11	x	0	1

Рис. 5. Словарь синхронных JK- и D-триггеров.

Dat и Pref (рис. 4) и далее будем вести синтез сразу для двух вариантов исполнения схемы: с использованием синхронных JK- и D-триггеров.

Теперь сделаем так называемый словарный перевод с «языка» изменений значений в парах кодирующих переменных (рис. 4) на «язык» значений инициирующих сигналов, вызвавших эти изменения. Словарь (рис. 5) устроен просто и представляет, по сути, таблицу истинности: преобразование текущего значения элемента памяти Q в последующее Q+ под действием комбинации управляющих сигналов, в нашем случае JK (для JK-синхронного триггера) или D (для D-синхронного триггера). Символ «x» озна-

чает безразличное значение, то есть либо «1», либо «0».

Выполнив переводы, получим карты Карно — Вейча с минимальными покрытиями мinterмами: управляющих сигналов для двух синхронных JK-триггеров (рис. 6а); для синхронного D-триггера (рис. 6б).

Выпишем оба варианта логических уравнений: (1) — для JK-триггеров, (2) — для D-триггеров, полученных по минимальным покрытиям на рис. 6, причем в некоторых случаях выгоднее выписывать инверсные значения переменных инициирующих сигналов:

$$\begin{aligned} \overline{J}_{pref} &= TxD \times \overline{Dat} + \overline{TxD} \times Dat; \\ K_{pref} &= J_{pref}; \\ J_{dat} &= TxD; \\ K_{dat} &= \overline{J}_{dat}; \\ \overline{D}_{pref} &= TxD \times \overline{Dat} \times \overline{pref} + \overline{TxD} \times pref \times \overline{Dat} + \\ &+ TxD \times pref \times Dat + \overline{TxD} \times \overline{pref} \times Dat; \\ D_{dat} &= TxD. \end{aligned} \quad (1)$$

$$\begin{aligned} \overline{D}_{pref} &= TxD \times \overline{Dat} \times \overline{pref} + \overline{TxD} \times pref \times \overline{Dat} + \\ &+ TxD \times pref \times Dat + \overline{TxD} \times \overline{pref} \times Dat; \\ D_{dat} &= TxD. \end{aligned} \quad (2)$$

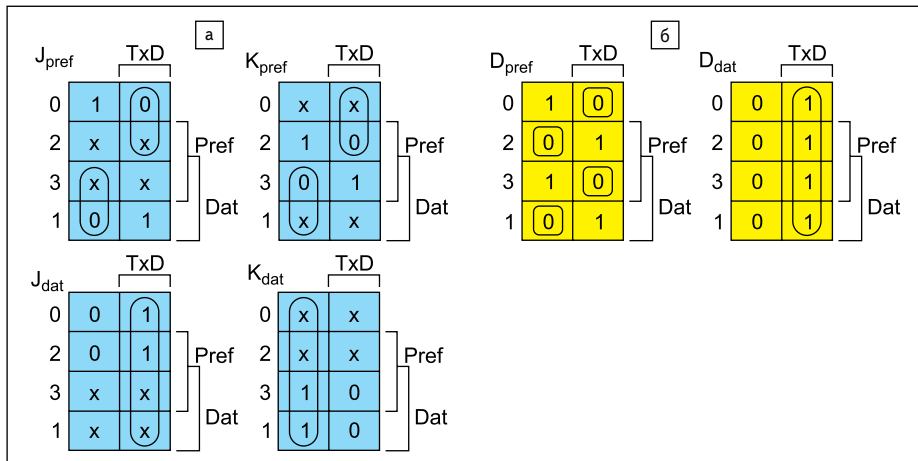


Рис. 6. Карты Карно — Вейча для: а) синхронного JK-триггера; б) синхронного D-триггера

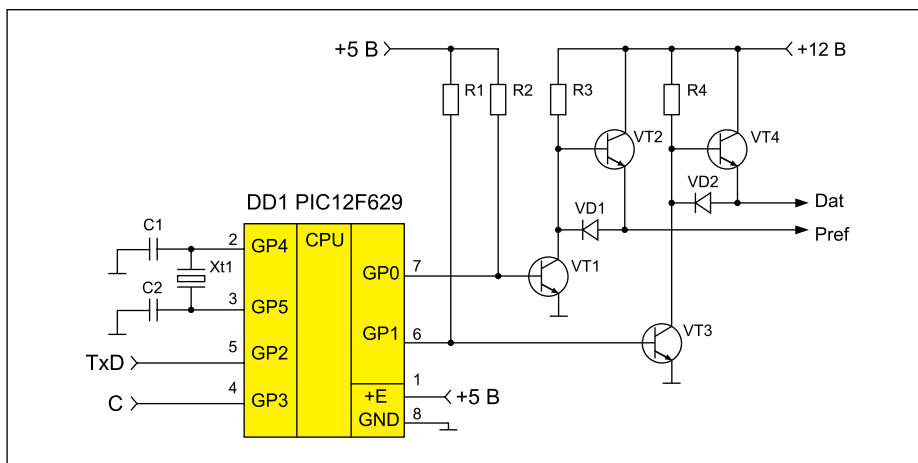


Рис. 7. Вариант синхронного автомата кодирования данных для последовательной передачи по телекоммуникационной линии на базе микроконтроллера PIC12F629 с драйвером двух сигнальных линий: Dat и Pref

### Выбор элементной базы для построения проекта

Итак, устройство должно представлять собой синхронный автомат, поэтому для его реализации выгодно использовать варианты с применением:

- микроконтроллера;
- синхронных триггеров из серий микросхем с малой степенью интеграции (МИС).

Но в обоих случаях нужно реализовать или синхронную таблицу переходов (рис. 3), или формулы (1), (2).

Проще всего реализовать синхронный автомат на микроконтроллере (рис. 7), выполняющем синхронную таблицу переходов (рис. 3), например, по следующему фрагменту программы, написанному на языке MicroPascal for PIC:

```

program PIC12F629;
var
  nbr:byte;
  {GP2=TxD, GP3=C, GP0=Pref, GP1=Dat}
procedure tab3;
begin
  begin
    case nbr of
      0:begin
        GPIO.GP0:=0; {Pref=0}
        GPIO.GP1:=0; {Dat=0}
        if GPIO.GP2=true
          then
            nbr:=1 {след. состояние автомата}
          else
            nbr:=2 {след. состояние автомата}
        end;
      1:begin
        GPIO.GP0:=0; {Pref=0}
        GPIO.GP1:=1; {Dat=0}
        if GPIO.GP2=true
          then
            nbr:=3 {след. состояние автомата}
          else
            nbr:=0; {след. состояние автомата}
        end;
      2:begin
        GPIO.GP0:=1; {Pref=0}
        GPIO.GP1:=0; {Dat=0}
        if GPIO.GP2=true
          then
            nbr:=3 {след. состояние автомата}
          else
            nbr:=0; {след. состояние автомата}
        end;
      3:begin
        GPIO.GP0:=1; {Pref=0}
        GPIO.GP1:=1; {Dat=0}
        if GPIO.GP2=true
          then
            nbr:=1 {след. состояние автомата}
          else
            nbr:=2; {след. состояние автомата}
        end
      else
        begin
          nbr:=0; {аварийный выход}
          GPIO.GP0:=0;
          GPIO.GP1:=0
        end;
    end;
  end;
begin
  TrisIO:=$0; {направление ввода-вывода}
  { Main program }
end.
    
```

На рис. 8 приведены синтезированные варианты схем синхронного автомата кодирования данных для последовательной передачи по телекоммуникационной линии с применением МИС: только JK-триггеров (рис. 8а, по формулам (1)) и только D-триггеров (рис. 8б, по формулам (2)).

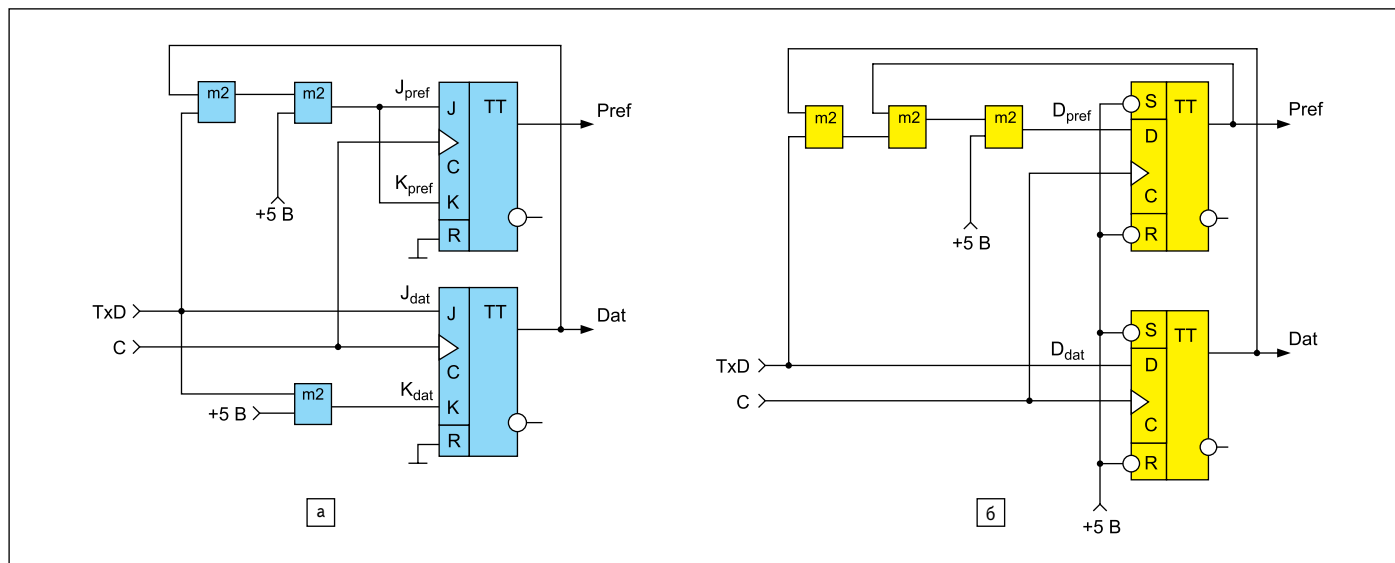


Рис. 8. Варианты логических схем синхронных автоматов кодирования данных для последовательной передачи по телекоммуникационной линии на базе: а) JK-триггеров; б) D-триггеров

На рис. 9 приведена еще одна схема на смешанной базе JK- и D-триггеров, построенная по формулам (1) для сигнала Pref и по формулам (2) для сигнала Dat. Она еще экономичнее: имеет на один элемент mod2 меньше.

Каждую из схем (на рис. 8, 9) следует дополнить драйвером для усиления сигналов Pref и Dat, аналогичным транзисторному драйверу на рис. 7.

#### Замечания по декодированию данных

В начале статьи на рис. 1 (графики 9 и 10) показаны временные диаграммы, иллюстрирующие работу синхронного автомата кодирования данных для последовательной передачи по телекоммуникационной линии, причем в линию связи передаются два сигнала — Dat и Pref.

Поскольку соседние переходы (рис. 2) закодированы кодом Грея, декодирование информации следует делать так:

- сначала восстановить сигнал синхронизации в виде суммы по модулю 2 кодирующих переменных ( $D \bmod 2$  (Pref));
- теперь по его и отрицательному, и положительному фронтам выделять данные из сигнала D.

Предложенный способ кодирования не обладает свойством сохранения постоянной составляющей в закодированной последовательности, но имеет свойство самопроверки пар соседних посылок по коду Грея.

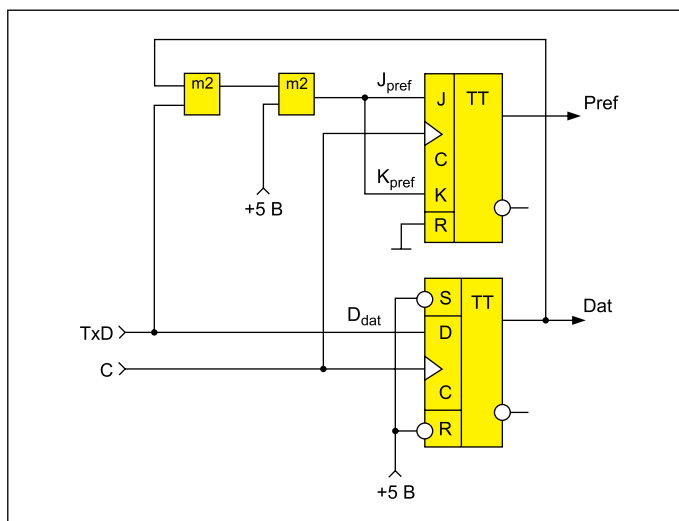


Рис. 9. Вариант логической схемы синхронного автомата кодирования данных для последовательной передачи по телекоммуникационной линии на смешанной базе

#### Литература

1. Гук М. Ю. Аппаратные интерфейсы ПК. Энциклопедия. СПб.: Питер, 2002.