Description

Communication interface for the serial transmission of digital data, and corresponding data transmission method

With serial data transmission, it is often necessary to make a compromise between the parameters of speed, interference immunity and current consumption. The highest data throughput at a given clock frequency is achieved by synchronous transmission using separate data, clock and control lines. However, such synchronous transmission is particularly susceptible to interference, and it has only conditional suitability for the use of data protection mechanisms for recognizing multiple errors.

Other methods, such as synchronous transmission with clock recovery, or asynchronous transmission, are significantly slower at the same clock frequency. The reason for this is the necessary transmission of additional information for synchronization and the multiple sampling required. The transmission rate can be increased by the proportionally rising current consumption only to a limited extent.

The object of the present invention is to specify a communication interface for the serial transmission of digital data and a serial data transmission method for the bitwise transmission of digital data, each of which can be used to achieve serial transmission which is immune to interference and has the speed advantages of synchronous transmission and reliable synchronization between the clock and the data.

It is already known practice to design the transmission path, by shielding or limiting the length, to be so reliable that recognition of individual errors is sufficient to ensure transmission with adequate interference immunity. In this case, the start and end of

data transmission are synchronized using start/stop synchronization via the control line. Bit synchronization is monitored by counting the clock pulses between start and stop. The data itself is protected by means of a parity bit.

In the case of synchronous transmission with clock recovery, the useful information is coded such that the resultant bit stream contains an adequate number of edge changes (Manchester coding, bit stuffing, 4B/5B etc.), which can be used to recover the send clock signal (bit synchronization). Start/stop synchronization is carried out using special bit sequences which do not occur in the rest of the telegram (BOF, EOF). The disadvantage of this is the relatively large volume of data to be transmitted as a result of coding (factor 1.25...2). In this context, the additional start and end identifier is of particular concern with small volumes of useful data.

With asynchronous transmission, bit synchronization is carried out by means of a start and stop bit. However, synchronization is assured only during a limited number of bit times, so that this sequence has to be repeated regularly. Start/stop synchronization of a telegram is carried out, as for synchronous transmission with clock recovery, with an explicit start and end identifier. The disadvantage here, too, is the relatively large volume of data to be transmitted. In addition, with asynchronous transmission, multiple sampling is necessary, which either reduces the data rate or increases current consumption.

The present invention solves the problem in that the information elements to be transmitted, the data, the clock signal and the start and end of data transmission (start/stop), are converted into special state sequences in the transmitter in accordance with a defined coding specification, and are transformed back from these state sequences into the aforementioned information elements again in the receiver.

To this end, a communication interface for the serial transmission of digital data is provided, in which at least three signal lines are provided which can each have a high or a low level impressed on them, and a data item which is to be transmitted can be coded by changing the level of two of the at least three signal lines (Tx0, Tx1, Tx2), and hence as a result of the transition from a first level n-tuple to a second level n-tuple.

In the case of exactly three signal lines (Tx0, Tx1, Tx2), the data item to be transmitted can be coded as a result of the transition from a first level triplet to a second level triplet. Since the communication interface can also be used to transmit synchronization information, for example, the term "data item" additionally at least includes such information as well.

The permissible transitions from a first level n-tuple or first level triplet to a second level n-tuple or second level triplet are defined in a coding scheme and can be stored in this form in the communication subscribers, which can be communicatively connected via the communication interface. The coding schemes in the receiver can therefore also be used, in particular, for error recognition and suppression.

This makes it possible to carry out a serial data transmission method for the bitwise transmission of digital data using at least three signal lines which can each have a high or a low level impressed on them, a data bit which is to be transmitted being coded by changing the level of two of the at least three signal lines (Tx0, Tx1, Tx2), and hence as a result of the transition from a first level triplet to a second level triplet.

In this case, data transmission is particularly susceptible to interference if the data bit which is to be transmitted is coded by a change in the level of two of the at least three signal lines.

If the change in the level of the at least two signal lines occurs in opposite senses, susceptibility to interference is reduced even further.

If the start and end of data transmission can be coded by inverting the respective levels of the at least three signal lines, start/stop synchronization with only one bit time is possible, so that, with short telegrams, a high data throughput is possible.

The invention and other advantageous refinements of the invention in accordance with features of the subclaims are explained in more detail below with the aid of schematically illustrated exemplary embodiments in the drawing, in which:

FIGURE 1 shows the principle of state sequence coding using a state diagram, and

FIGURE 2 shows the transmission of useful data with the state sequence coding.

In accordance with a defined coding specification, which is explained below with reference to Figure 1, the information elements to be transmitted are converted into special state sequences.

The state diagram shown in Figure 1 has six nodes with transitions defined between these nodes. Each of the nodes represents a level, status or state triplet. During bus transmission, only the states 001_B, 010_B and 100_B are used, because, with these

state triplets, exactly two levels change in opposite senses whenever there is a transition from a first to a second triplet. Under these circumstances, the transition from the level triplet 001_B to the level triplet 010_B identifies the transmission of a logic 1; the transition from the level triplet 001_B to the level triplet 100_B identifies the transmission of a logic 0 etc.

During bus inactivity R, only the states 110_B, 101_B and 011_B are used, since, again, exactly two levels change in opposite senses during transition from a first state triplet of this quantity to a second state triplet of this quantity.

To change from the operating mode of data transmission T to the operating mode of bus inactivity R or to change from the operating mode of bus inactivity R to the operating mode of data transmission T, a start/stop transition is necessary. This start/stop transition is distinguished by inversion of all three levels. If, in the last step of data transmission, a logic 1 is transmitted with the change from the level triplet 010_B to the level triplet 100_B, there follows a stop transition from the level triplet 100_B to the level triplet 011_B. Thereupon comes the transition to the operating mode of bus inactivity R, in which the levels associated with the level triplet 110_B are applied to the three signal lines. From this state of bus inactivity R, a start transition is used to change to the operating mode of data transmission T - level triplet 001_B - again, a logic 1 being identified by a transition to the level triplet 010_B and a logic 0 being identified by a transition to the level triplet 100_B.

The present state of the three signal lines is denoted by a respective node on the state diagram. At a particular instant, only one of the nodes on the state diagram is ever valid. The nodes on the state diagram describe, as it were, positions on a path, the

transitions defined between the nodes stipulating the possible paths. Transitions not defined on the state diagram are not possible; thus, Figure 1, direct transition from the level triplet 001_B to the level triplet 011_B, for example, is not possible. The coding specification defined using the state diagram shown in Figure 1 is called a state sequence coding. In addition, it is also possible to use data protection mechanisms for recognizing multiple errors and for error correction.

Figure 2 shows an example of data transmission based on the principle of state sequence coding. Both the transmitter S (first ordinate in the upper third) and the receiver E (first ordinate in the lower third) contain the data D, D' in bit-serial form. The data D, D' is distinguished conventionally by alternate high and low levels, a plurality of successive similar levels being distinguishable by means of the clock signal C (central ordinate, both in the upper third and in the lower third). Data transmission starts when a high level is impressed on the start/stop line and ends when the level of this line changes to low again (third ordinate both in the upper and in the lower third).

The central third shows the signal sequence on the transmission path L, the three signal lines (Tx0, Tx1, Tx2), which is produced after the transmission data D has been subjected to state sequence coding (first ordinate, upper third). The transmission data D is converted into bit-serial data D' again in the receiver E in accordance with the coding specification of the state sequence coding (first ordinate, lower third).

* C L M S *

Patent Claims

1. A communication interface for the serial transmission of digital data, in which at least three signal lines (Tx0, Tx1, Tx2) are provided which can each have a high or a low level impressed on them, and a data item which is to be transmitted can be coded by changing the level of at least two of the at least three signal lines (Tx0, Tx1, Tx2), and hence as a result of the transition from a first level n-tuple to a second level n-tuple.

2. The communication interface as claimed in claim 1, characterized in that the first level n-tuple and the second level n-tuple are a first level triplet and a second level triplet.

3. The communication interface as claimed in claim 1 or 2, characterized in that the level change on the at least two signal lines occurs in opposite senses.

4. The communication interface as claimed in claim 1, 2 or 3, characterized in that the start and the end of data transmission can be coded by inverting the respective levels of the three signal lines (Tx0, Tx1, Tx2).

5. A serial data transmission method for the bitwise transmission of digital data, in which at least three signal lines (Tx0, Tx1, Tx2) are provided which can each have a high or a low level impressed on them, and a data item which is to be transmitted can be coded by changing the level of two of the at least three signal lines (Tx0, Tx1, Tx2), and hence as a result of the transition from a first level n-tuple to a second level n-tuple.

* C L M S *

6. The data transmission method as claimed in claim 5, characterized in that the first level n-tuple and the second level n-tuple are a first level triplet and a second level triplet.

7. The data transmission method as claimed in claim 4 or 5, characterized in that the start and the end of data transmission is coded by inverting the respective levels of the three signal lines (Tx0, Tx1, Tx2).

8. An automation appliance for controlling and/or monitoring a technical process, which can be communicatively connected to other automation appliances, characterized in that the automation appliance has a communication interface as claimed in claim 1, 2, 3 or 5.

9. An automation appliance for controlling and/or monitoring a technical process, which can be communicatively connected to other automation appliances, characterized in that data interchange between the communicatively connected automation appliances takes place using a data transmission method as claimed in one of claims 5, 6, 7 and 8.