

received by the communication device 140 is stored, and is always kept up-to-date.

The communication data database 131 may store not only received data but also transmitted data.

[0028] The consistency information storage unit 230 is configured by the read ID

5 management table 132, the write ID management table 133 and the consistency information database 134 stored in the RAM 130. The read ID management table 132 is a management table of read IDs assigned for respective data reading processes from the communication data database 131 performed by the application block 201, and associated with read target data of respective read processes. The write ID

10 management table 133 is a management table of write IDs assigned for respective data writing processes into the communication data database 131 performed by the device driver 203, and associated with write target data of respective writing processes.

Further, the consistency information database 134 is a database to store consistency information for judging consistency of data read from the communication data database

15 131 by the application block 201.

[0029] The data writing processing unit 240 is configured by a write start process 241, a write matching start process 242, a writing process 243, a write matching process 244, a write finish process 245 and a write matching finish process 246, which are called from the device driver 203 at reception timing specified by a communication protocol.

20 Here, respective processes configuring the data writing processing unit 240 correspond to means configuring the embedded system 100. The write start process 241 is called from the device driver 203, and starts writing processing of received data into the communication control unit 202. The write matching start process 242 is called from the write start process 241, and updates the consistency information storage unit 230 to

25 a state indicating start of data writing. The writing process 243 is a process to write

The assignment method of read IDs will be described below in detail.

[0037] The value of the read ID number is copied to the writing-time ID number at the time when the write matching process 244 is performed, and the writing-time ID number becomes 0 in a case wherein the read ID number becomes 0 at the time when the read matching finish process 215 is performed. The initial value is 0.

[0038] Next, operation will be discussed.

First, an overview will be given using Fig. 2. First, regarding the communication data database 131, operation of the data writing processing unit 240 is given priority over the operation of the data reading processing unit 210. For realizing the same, the data reading processing unit 210 and the data writing processing unit 240 are operated independently, and in a case wherein writing of received data is performed by the operation of the data writing processing unit 240 while data reading is in progress by the operation of the data reading processing unit 210, writing by the data writing processing unit 240 is regarded as success in writing, and reading by the data reading processing unit 210 is regarded as failure in reading since data read is inconsistent. At the time of reading failure, it may be applicable to make the data reading processing unit 210 be retried and repeated until received data read obtains consistency. If it is impossible to acquire received data having consistency even with a fixed number of times of retry, reading failure is notified to a caller of the data reading processing unit 210 as error processing. Hereinafter, a case wherein retry is performed will be described as an example.

[0039] Judgment of success or failure in reading is performed by read IDs. The read IDs are managed in the read ID management table 132 inside the consistency information storage unit 230, and are assigned at the time when a read matching start process 212 is performed. The read IDs once assigned become unassigned when a

S201). In this way, it is managed that preparation to write is made.

[0048] In a step S202, a new write ID is assigned to a writing process started in the write matching start process 242, and the new write ID is registered in the write ID management table 133 in the consistency information storage unit 230. An example of the management table of write IDs assigned is illustrated in Fig. 8. Specifically, a write ID is assigned to a set of one or more pieces of target data for which a writing request has been made from the device driver 203, and as the value, a value after incrementing the write-start count is assigned.

In this manner, since a write ID is assigned to a writing process, it is possible to manage a plurality of pieces of target data written in one writing process as one set.

[0049] Next, storage locations inside the communication data database 131 of write target data is retrieved from the communication data management table 124 in the writing process 243 (step S203), and received data is written based on the location information (step S204). Next, information inside the consistency information database 134 is updated in the write matching process 244. Specifically, a data writing-in-progress flag is made to be TRUE, and the read ID number is copied to the writing time ID number (step S205). In this manner, all the data reading by using the read IDs already assigned is judged as not having consistency at the time when the read matching finish process 215 is performed.

[0050] In a step S206, it is judged all the pieces of write target data which is requested from the device driver 203 in the write finish process 245 have been written or not. If there is remaining write target data, procedure returns to the step S203, the step S204 and the step S205, and target data is further written. When all data of write target is written, procedure is shifted to a step S207. Necessary data here is a set of target data for which a writing request has been made from the device driver 203, and is judged by

by the read matching finish process 215. Since the data writing in-progress flag is FALSE, and the read ID assigned is 1 while the writing time ID number is 0, hence the value of the read ID is larger, it is judged that consistency exists. After judgment, the contents of the consistency information becomes (D). Next, a writing process is

5 performed, and the contents of the consistency information is updated to (E), (F) and (G) in accordance with the rule as described above. In the next (H), the read matching finish process 215 is performed as in (D), wherein whether consistency exists is judged, and a data writing-in-progress flag and a writing-time ID number are referred to in (G). Here, since the data writing-in-progress flag becomes FALSE, and the read ID becomes

10 2 while the writing time ID number becomes 2, hence the read ID and the writing time ID number are the same, and the condition for consistency presence is not fulfilled, it is judged that consistency does not exist. After judgment, the contents of the consistency information becomes (H).

[0057] As results of judgment at the time when the read finish processes 214 in (D) and (H) are performed as described above, reading taking (B) through (D) as a set is judged to have consistency, and reading taking (C) through (H) as a set is judged not to have consistency, which are judgment results as expected.

[0058] Explanation is provided of an example in a case wherein start of writing is included in the period from start of reading to finish, using Fig. 10. In this example,

20 since writing is actually performed after (D) although writing is started between (B) through (D), it is expected that consistency is judged to exist.

[0059] When consistency information inside the consistency information database 134 is updated in accordance with the rule as described above, in the consistency information of (C) referenced at the time of consistency judgment when reading is

25 finished in (D), the data writing-in-progress flag becomes FALSE, and the read ID

assigned becomes 1 while the writing time ID number becomes 0, hence the value of the read ID becomes larger; therefore, it is judged that there is consistency in reading. This result is a judgment result as expected.

[0060] Explanation is provided of an example of a case wherein writing is included in the period from start to finish of reading, using Fig. 11. In this example, since writing is performed in the period from start to finish of reading, it is expected that consistency is judged not to exist.

[0061] When consistency information inside the consistency information database 134 is updated in accordance with the rule as described above, in the consistency information of (D) referenced at the time of judging consistency in (E), the data writing-in-progress flag becomes TRUE, and the read ID assigned becomes 1 while the writing time ID number becomes 1; thus, since the data writing-in-progress flag is not FALSE, and further, the read ID and the writing time ID number are the same, and the condition for consistency presence is not fulfilled, it is judged that consistency does not exist. This result is a judgment result as expected.

[0062] Explanation is provided of an example of a case wherein writing finish is included in the period from start to finish of reading, using Fig. 12. In this example, since writing is finished without performing writing during the period until finish of reading, although writing is started at the start point of reading, it is expected that consistency is judged to exist.

[0063] When consistency information inside the consistency information database 134 is updated in accordance with the rule as described above, in the consistency information of (E) referenced at the time of consistency judgment in (F), the data writing-in-progress flag becomes FALSE, and the read ID assigned becomes 1 while the writing time ID number becomes 0, hence the value of the read ID becomes larger;

[0072] Explanation is provided of an example of a case wherein reading taking (B) through (G) as a set, reading taking (E) through (H) as a set, and writing taking (C), (D) and (F) as a set are performed, and writing is performed at a timing between (B) and (E), using Fig. 17. In this example, it is expected that reading between (B) and (G) is
 5 judged not to have consistency since data is written between (B) and (G), and reading between (E) and (H) is judged to have consistency since writing is not performed and writing is finished between (E) and (H) until finish of reading.

[0073] When consistency information inside the consistency information database 134 is updated in accordance with the rule as described above, in the consistency
 10 information of (F) referenced at the time of consistency judgment in (G), the data writing-in-progress flag becomes FALSE, and the read ID assigned becomes 1 while the writing time ID number becomes 1, hence the read ID and the writing time ID number are the same, and conditions for consistency presence are not fulfilled; therefore, it is judged that consistency does not exist. Further, in consistency information of (G)
 15 referenced at the time of consistency judgment in (H), the data writing-in-progress flag becomes FALSE, and the read ID assigned becomes 2 while the writing time ID number becomes 1, hence the value of the read ID is larger; therefore, it is judged that consistency exists.

[0074] As described above, as results of judgement at the execution time of the read
 20 finish processes 214 of (D) and (H), reading taking (B) through (G) as a set is judged not to have consistency, and reading taking (E) through (H) as a set is judged to have consistency, which are judgment results as expected.

[0075] Explanation is provided of an example of a case wherein start of writing and finish of writing are included, and further, writing is included for two times, in the
 25 period from start to finish of reading, using Fig. 18. In this example, since writing is

performed between (B) and (G), it is expected that consistency is judged not to exist.

[0076] When consistency information inside the consistency information database 134 is updated in accordance with the rule as described above, since writing is performed for two times in (D) and (E), the data writing-in-progress flags and the writing time ID

5 numbers are updated in each. In the consistency information of (F) referenced at the time of consistency judgement in (G), the data writing-in-progress flag becomes FALSE, and the read ID assigned becomes 1 while the writing time ID number becomes 1, hence the read ID and the writing time ID number are the same, and conditions for consistency presence are not fulfilled; therefore, it is judged that consistency does not exist.

10 [0077] As described above, in any examples, when received data is acquired from the application block 201, it is possible to judge consistency of the received data. When there is no consistency, by retrying reading, and repeating reading until data having consistency is read, it is possible to ensure that the data read has consistency. Further, since writing of received data from the device driver 203 is made to precede a reading
15 process of received data from the application block 201, the processing of the device driver 203 is prevented from waiting the processing of the application block 201, and it becomes easy for the processing of the device driver 203 to ensure the timing of the processing specified by a communication device and a communication protocol.

In the description above, explanation is provided of the case wherein data
20 received by a communication device is read by an application block; however, the present invention is not limited to this. For example, it is also applicable to a case wherein data obtained by several kinds of sensors is recorded in a memory, and the data is used by an application block, and so on.

[0078] Further, the consistency information database 134 added for judgment of
25 consistency presence or absence in the embodiment of the present invention is small in

size, which does not duplex communication data nor degrades memory use efficiency as in the technique illustrated in Patent Literature 2. Therefore, it is possible to easily apply the consistency information database 134 to a product or a system including a small-scale memory which does not allow duplex of communication data.

5 [0079] As described above, since the embedded system 100 of the first embodiment is configured to judge consistency presence or absence of a reading process taking one or more pieces of data as targets, it is possible to ensure that consistency exists in a plurality of data read, and to obtain an effect that control without any malfunction is made possible, according to the embedded system 100.

10 [0080] Second Embodiment

In the first embodiment, explanation is provided of the embodiment wherein when writing is performed between start and finish of reading, it is judged failure in reading in all cases, and retry is required. In a second embodiment, explanation is provided of an embodiment which is possible to judge consistency presence or absence
15 more suitably by judging failure in reading only when data relevant to consistency of data read is written.

[0081] Specifically, it is realized by classifying data inside the communication data database 131 into several groups relative to consistency, and in a case wherein data being a reading target requested by the application block 201 and data being a writing
20 target requested by the device driver 203 belong to different groups, by not judging failure in reading even reading or writing occurs simultaneously. As a result, it is possible to reduce the occurrence frequency of reading retry.

[0082] Fig. 19 is a configuration diagram illustrating one example of a system configuration of an embedded system 300 according to the second embodiment of the
25 present invention, and Fig. 20 is a configuration diagram illustrating one example of a

software configuration of software 400 of an embedded system according to the second embodiment of the present invention.

[0083] In Fig. 19, the embedded system 300 is different from the embedded system 100 of the first embodiment in that a ROM 120 stores a group classification

5 management table 125, and the other components are the same.

[0084] Further, in Fig. 20, the software 400 of the embedded system is different from the software 200 of the embedded system of the first embodiment in that a consistency information storage unit 230 is equipped with a group classification management table 125, and the other components are the same.

10 [0085] The group classification management table 125 is a management table of a group wherein data inside the communication data database 131 is classified into data relative to consistency, and an example of the data structure is illustrated in Fig. 21.

The embedded system 300 specifies a group to be a target, by referencing the group classification management table 125 at the time of performing a read start process 211

15 by the data reading processing unit 210 and a write start process 241 by the data writing process unit 240, and when the consistency information database 134 is referenced or updated in the following processing, the group specified is used. Accordingly, if groups to be targets are different, even when reading or writing occurs simultaneously, the groups of consistency information used for consistency judgment are different;

20 hence, it is not judged failure in reading.

[0086] Further, Fig. 22 is one example of a data structure of consistency information inside the consistency information database 134 in the second embodiment, and the configurations of the consistency information of respective groups are the same as those in the first embodiment as illustrated in Fig. 4. Fig. 21 and Fig. 22 present cases as

25 examples wherein a whole of the communication data is divided into four groups;

meanwhile, the number of groups to be divided into can be arbitrarily selected.

Further, as for a classification of groups, it may be applicable to classify for every product or system being an object such as a room air conditioner, a television, etc., or for every process being an object, such as an air conditioning setting change process,
 5 etc.

Since the components other than the group classification management table 125 in Fig. 19 and Fig. 20 are the same as those in Fig. 1 and Fig. 2 of the first embodiment, for which detailed explanation is omitted.

[0087] Explanation is provided of a processing sequence in the data reading
 10 processing unit 210 in the second embodiment, using Fig. 23.

The difference from Fig. 5 where communication data is not divided into groups is that a process to select a group being a reading target from consistency information illustrated in Fig. 22 (step S301) is added, and the other processes (from a step S302 through a step S312) are processes corresponding to the step S101 through
 15 the step S111 in Fig. 5, respectively. However, in the processes related to consistency information (a step S304, a step S308 and a step S309), consistency information of the group selected in the step S301 is used.

[0088] First, a group being a reading target is selected in a read start process 211 (step S301). Specifically, by referencing Fig. 21, an object group of data for which a
 20 reading request has been made from the application block 201 is specified, and by referencing Fig. 22, consistency information of the object group is selected. The object group in the consistency information selected is referenced or updated in the following processing. Next, procedure proceeds to a process to make the retry count be 0 (step S302), a process to confirm whether the retry count reaches the number of
 25 times determined beforehand (step S303), and an error process at the time of reading

failure when the retry count reaches N times being the maximum retry count set beforehand (step S312). Since these processes are processes corresponding to the steps S101, S102 and S111 in Fig. 5, the detailed explanation is omitted.

[0089] When the retry count does not reach N times being the maximum retry count in the step S303, procedure proceeds to the step S304. In the step S304, a new read ID is assigned to a reading process started, in the read matching start process 212, and is registered in the read ID management table 132 of the consistency information storage unit 230. Here, the reading-in-progress ID number is also incremented. Since this processing is processing corresponding to the step S103 in Fig. 5, the detailed explanation is omitted.

As for the read ID, one ID is assigned to reading target data belonging to a same group. If it is necessary to read data of a plurality of groups simultaneously, a read ID is assigned for each group, and is respectively managed separately.

[0090] Next, storage locations inside the communication data database 131 of reading target data is retrieved from the communication data management table 124 in the reading process 213 (step S305), and received data is read based on the location information (step S306). Here, in data reading, only reading target data belonging to a same group is read. If it is necessary to read data of a plurality of groups, the present processing sequence is performed for each group.

[0091] In a step S307, it is judged whether all data being a reading target with the same read ID in the same group is read in the read finish process 214. Since this process is a process corresponding to the step S106 in Fig. 5, the detailed explanation is omitted.

[0092] The judgment process of consistency presence or absence in the step S308 and the process to make a read ID be unassigned in the step S309 are processes

corresponding to the step S107 and the step S108 in Fig. 5, respectively. Therefore, the detailed explanation of the processing is omitted. However, the consistency information referenced here is consistency information of the group selected in S301.

[0093] In a judgment processing step S310, a process to proceed to is judged by the consistency presence or absence judged in the step S308. If it is judged that consistency does not exist in the step S310, the retry count is increased (S311) by 1, and procedure returns to judgment in the step S303. If it is judged that consistency exists in the step S310, procedure is finished deeming reading successful.

The processing sequence in the data reading processing unit 210 is described in Fig. 23; however, the processing sequence is not limited to the description. For example, if it is judged that consistency does not exist in the step S310, procedure returns to the step S303, and reading is performed again; however, it may be applicable to adopt a processing sequence wherein the processing regarding retry is omitted, and procedure is shifted to the step S312 deeming reading unsuccessful.

[0094] A processing sequence in the data writing processing unit 240 according to the second embodiment will be described using Fig. 24.

The difference from Fig. 7 wherein communication data is not divided into groups is that a process to select a group being a writing target from the consistency information illustrated in Fig. 22 is added (step S401), and the other processes (from a step S402 through a step S407) are processes corresponding to the step S201 through the step S206 in Fig. 7, respectively. However, in the processes (the step S402, the step S405 and the step S407) relative to consistency information, information in the line selected in the step S401 is used.

[0095] First, consistency information of a group being a writing target is selected in the write start process 241 (step S401). Specifically, a target group of data for which a

writing request has been made from the device driver 203 is specified by referring to Fig. 21, and by referring to Fig. 22, consistency information of the target group is selected. Here, the target group in the consistency information selected is referenced or updated in the following processing.

5 [0096] Next, procedure proceeds to a process to record start of received data writing in consistency information in the consistency information database 134, in the write matching start process 242 (step S402), a process to assign a new write ID to the write processing started, and to register the new write ID in the write ID management table 133 of the consistency information storage unit 230 (step S403), a process to read
10 storage locations inside the communication data database 131 of the write target data from the communication data management table 124, in the writing process 243 (step S404), and a process to write received data based on the locations (step S405). These processes are processes corresponding to the steps S201, S202, S203 and S204 in Fig 7, and the detailed description is omitted. Meanwhile, as for the write ID, one ID is
15 assigned to write target data belonging to a same group. If it is necessary to read data of a plurality of groups simultaneously, write IDs are assigned to respective group, which are managed separately, and the present processing sequence is performed for respective groups.

[0097] Next, the consistency information inside the consistency information database
20 134 is updated in the write matching process 244 (step S406). The process is the process corresponding to the step S205 in Fig. 7, and the detailed description is omitted. Here, the consistency information to be updated here is consistency information of the group selected in the step S401.

[0098] In a step S407, it is judged whether all data being a writing target which is
25 requested from the device driver 203 in the write finish process 245, and in the step

S408, information on the start of received data writing recorded in the consistency information database 134 in the step S402 is deleted, and data writing is finished deeming writing successful. These processes are the processes corresponding to the step S206 and the step S207 in Fig. 7, and the detailed description is omitted.

5 In the processing sequence of the data writing processing unit 240, timing to update consistency information regarding data writing is not limited to the timing illustrated in Fig. 24. For example, the step S405 to reflect data writing on the consistency information may be performed before the write step S404 of received data.

[0099] As described above, according to the embedded system 100 of the second
10 embodiment, data inside the communication data database 131 is divided into groups relative to consistency judgment, and consistency of read processing taking one or more pieces of data as a target is judged for respective groups; therefore, even in a case wherein reading from the application block 201 and writing from the device driver 203 are performed at the same time, it is possible to make an effect to have consistency be
15 judged to exist when read target data requested from the application block 201 and received data requested from the device driver 203 do not belong to a same group, to suitably judge that consistency exists in a plurality of pieces of data on which the reading process has been performed, and to obtain an effect to make control without any malfunction possible. It is also possible to reduce the possibility of performing
20 redundant processing such as repeating retry, and to improve processing efficiently.

Reference Signs List

[0100] 100, 300: embedded system; 110: CPU; 120: ROM; 121: application block code; 122: communication control unit code; 123: device driver code; 124: communication data management table; 125: group classification management table;
25 130: RAM; 131: communication data database; 132: read ID management table; 133:

write ID management table; 134: consistency information database; 140:
communication device; 200, 400: software; 201: application block; 202: communication
control unit; 203: device driver; 210: data reading processing unit; 211: read start
process; 212: read matching start process; 213: reading process; 214: read finish
5 process; 215: read matching finish process; 220: communication data storage unit; 230:
consistency information storage unit; 240: data writing processing unit; 241: write start
process; 242: write matching start process; 243: writing process; 244: write matching
process; 245: write finish process; 246: write matching finish process

CLAIMS

[Claim 1] An embedded system comprising:

a read start means to start a reading process of data performed by an application

5 unit;

a read matching start means to add one to a read ID number indicating the number of reading processes being executed when the reading process is started by the read start means, and to assign a value after adding one as a read ID of the reading process started by the read start means;

10 a reading means to read the data;

a read finish means to finish reading of the data by the reading means and to finish the reading process; and

a read matching finish means to acquire the read ID assigned to the reading process finished by the read finish means, a data writing-in-progress flag to indicate whether a writing process of data into an area from which the data is read by the reading process is in progress at the time of finishing the reading process by the read finish means, and a writing time ID number indicating the read ID number at the time of starting the writing process, and when the data writing-in-progress flag indicates that the writing process is not in progress and the acquired read ID is a value larger than the writing time ID number, to judge that consistency exists in the data read, and to subtract one from the read ID number of the time of finishing the reading process by the read finish means.

[Claim 2] The embedded system as defined in claim 1,

wherein the read matching finish means performs a data reading process again

25 when the read matching finish means judges that consistency does not exist in the data

read.

[Claim 3] The embedded system as defined in claim 1 or claim 2,

wherein the read ID, the read ID number, the data writing-in-progress flag and the writing time ID number are divided into a plurality of groups for each data being an object, and

wherein the read matching finish means judges consistency for each of the plurality of groups.

[Claim 4] An embedded system control method to control a data reading process from a shared memory of an embedded system equipped with the shared memory, the embedded system control method comprising:

a read ID assignment step to add one to a read ID number indicating the number of reading processes being executed at the time of starting the data reading process, and to assign a value after adding one as a read ID of the reading process started, by a read matching start means;

a data reading step to read data from the shared memory, by a reading means;

a consistency information acquisition step to acquire, at the time of finishing the data reading step, consistency information including the read ID number assigned to the reading process finished, a data writing-in-progress flag to indicate whether a data writing process into an area in the shared memory from which the data is read by the reading process is in progress at the time of finishing the reading process, and a writing time ID number indicating the read ID number at the time of starting the writing process, by a read matching finish means;

a consistency judgment step to judge that consistency exists in the data read in the data reading step when the data writing-in-progress flag acquired in the consistency information acquisition step indicates that the writing process is not in progress, and the

read ID acquired in the consistency information acquisition step is larger than the writing time ID number, by the read matching finish means; and

a matching judgment finish step to subtract one from the read ID number of the time of finishing the reading process by the reading means, by the read matching finish

5 means.