ON Semiconductor

Is Now

To learn more about onsemi™, please visit our website at
www.onsemi.com

AND9761

Serial Communication Protocol SWAN for the Fan Driver Product Family

ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

INTRODUCTION

Serial interface to access the built−in setting and control registers of the motor driver product family, has been provided. It is utilized by multiplexing a simple conventional interface; FG/RD and PWM signal ports which most fan units have. This application note describes how to use this communication feature.

OVERVIEW

This communication feature is named SWAN (Smart Wire Access Network). The physical layer of SWAN is:

- Compatible with UART
- Selectable from either single wire or dual wire mode
- Connected through pin FG for single wire or pins FG and PWM for dual wire.

Detail is described in the section “Connection“.

The communication mode and normal operation mode are completely separated. The device must work as usual motor driver by just power−on. The special protocol is required to activate SWAN, whose details are described in the section “Operation“.

The data format of SWAN consists of:

- Header
- Register address
- Data length in byte
- Parity
- Data in byte
- Check−sum

Details are described in the section “Data Format“.

CONNECTION

Figure 1 and Figure 2 show the connection between the device and MPU. Figure 1 is for single wire mode and Figure 2 is dual wire mode. FG pin must be pulled−up through a resistor on the PCB. In single wire mode, FG pin must be connected to an open−drain bidirectional port of an MPU, and PWM pin is not used for serial communication. In dual wire mode, PWM pin must be connected to an open−drain (with pull−up resistor) or sink−source output port of an MPU. The pin VDD is recommended for the pull−up voltage source. Port selection and setting must follow MPU instruction, if needed. An MPU with UART support might be useful.

TSL pin is used to determine single or dual wire mode. Logical low is single wire mode, and logical high is dual wire mode.

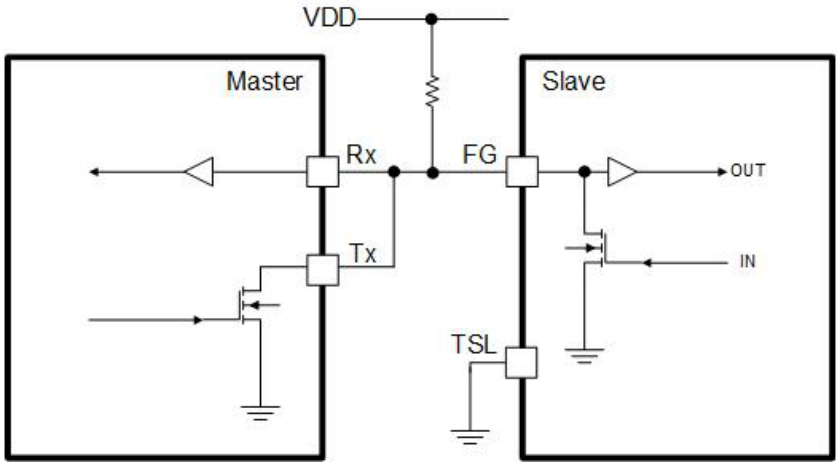

Figure 1. Single wire mode connection

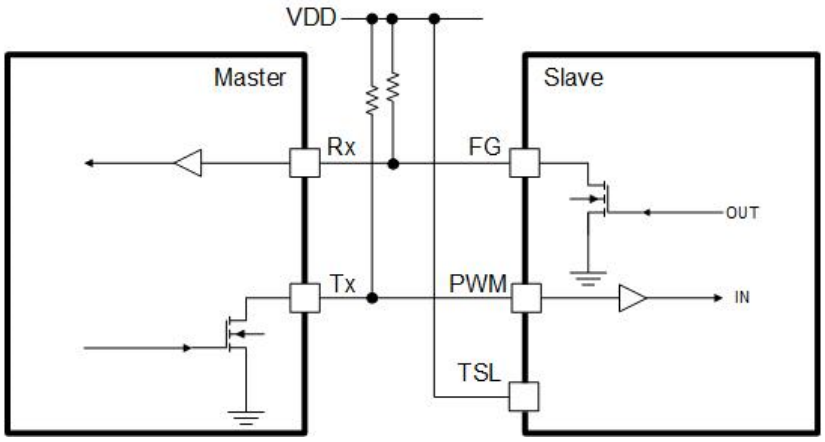

Figure 2. Dual wire mode connection

OPERATION

The followings are the procedure to switch to SWAN mode. The detail of the data format, “Field“ and “Frame“ are explained later. In this section, the MPU which controls the communication is called “Master“, and the motor driver is called “Slave“. The following procedure must be done by the Master device.

1. Power−off the fan.
2. Connect FG (and PWM) to the Master while TSL pin is set properly in advance.
3. Power−on the fan.
4. Start sending the special pattern “Header Field“ from the Master through UART Tx port to the Slave within 1 ms after power−on.
5. Send the “Header Field“ repeatedly for appropriate times described below.
6. Send “Data Frame(s)“ to manipulate the registers.

The Slave works as follows:

1. To determine normal mode or “Communication“ mode, at 1ms later from power-on, the Slave starts checking if the special UART pattern is input.
2. While “Header Field“ is repeatedly input, the Slave detects it and fixes the baud rate automatically. The minimum number of times the Header Field repeats depends on the baud rate, and it is shown as following eq.1.

$$N = \frac{(24/14) \times R}{11} + 9 \qquad \text{(eq. 1)}$$

where
N: minimum number the header field repeats [count]
R: Baud rate [kpbs]

After fixing the baud rate, the Slave goes to “Power−on Standby“ mode and is ready to access the

internal register. The motor doesn't rotate in this mode and the Slave is waiting for the next signal or data. If the Slave fails to detect "Header Field", the baud rate is not fixed and the preparation for serial communication fails. Then, the Slave goes to "Motor drive" mode (i.e. normal mode) and the motor rotates.

3. After entering "Power−on Standby" mode, if the Slave detects "Read/Write Field" within 9% of 1 field transfer time, it goes to "Communication" mode. If the Slave doesn't detect it, it goes to "Standby" mode and waits for a next Frame. Refer Figure 3. State transition diagram of the communication.
4. In the communication mode, using the remaining Fields, the Master data is written to the target register of the Slave as write operation, if the check−sum value is correct. As for read operation, the registered value read from the target register is sent to the Master.
5. When the above communication process is finished successfully, the Slave goes to "Standby" mode. In this mode, the signal pin should be kept high to prevent the communication error. To return back to "Communication" mode again from this mode, a new "Frame" input is needed.

Figure 3 shows the state transition diagram of SWAN.

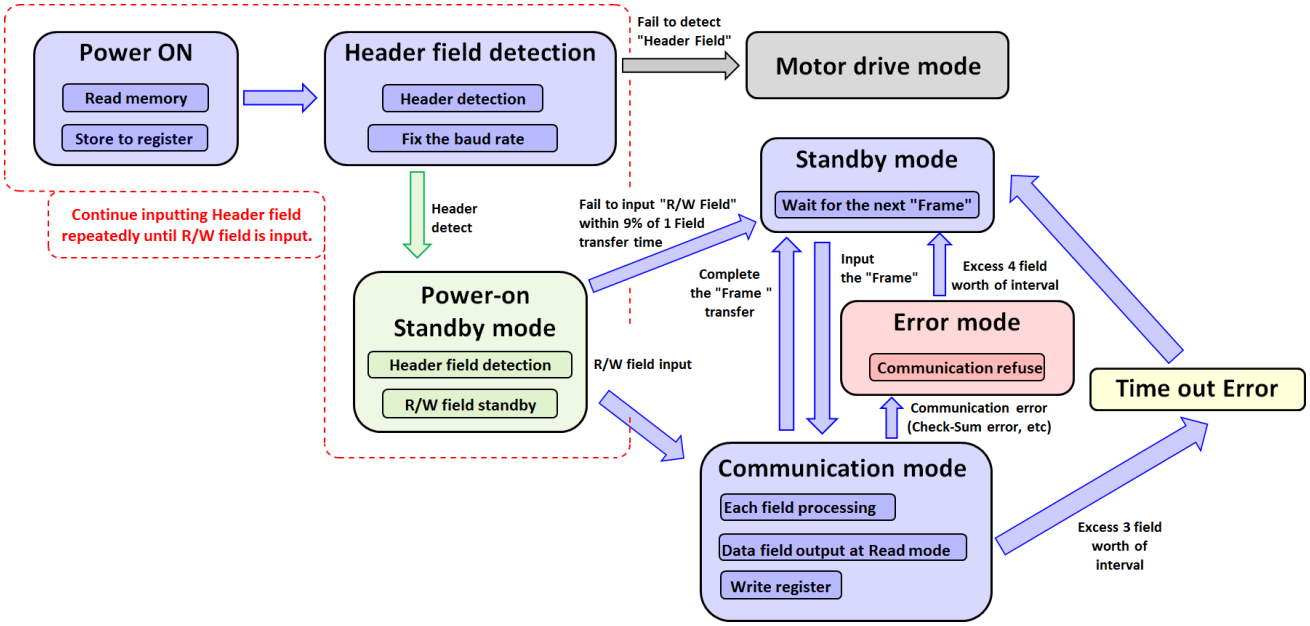

Figure 3. State Transition Diagram of the Communication

SIGNAL CONDITION

The data transfer speed is supported from 2.4 kbps to 400 kbps. The Slave automatically calculates the baud rate from the "Header Field". Therefore the transfer speed change during the communication causes the communication error. The interval between Fields is allowed 3 Fields maximum.

The low state of the communication pin between Fields is recognized as a false start bit and causes communication error. Therefore, it is important to keep it high.

DATA FORMAT

SWAN communication commands follow the format as below.

"Field" Format

Figure 4 shows the data format called "Field" and it is basic of data transfer. It consists of 11 bits and is 8N2 format, in which there are 1 start bit, 8 data bits (LSB first) and 2 stop bits. The communication starts when falling edge of the start bit is detected and ends when the rising edge of last stop bit is detected. Stop bit is fixed high.

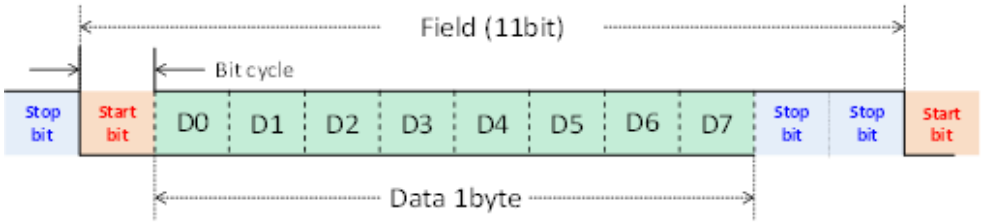

Figure 4. Field Format

"Frame" Format

Figure 5 shows "Frame" format in write mode which consists of at least 7 Fields. It is unit of the communication and starts with "Header Field" and ends with "Check−Sum Field". If the data length is long, one frame contains several checksum fields. Regardless of the length of the frame, the frame ends with the last checksum field.

There is no special signal or gap between each Field. After transferring the Frame successfully, the communication pin stays high and the IC waits for next start bit. This state is called "Standby" mode.

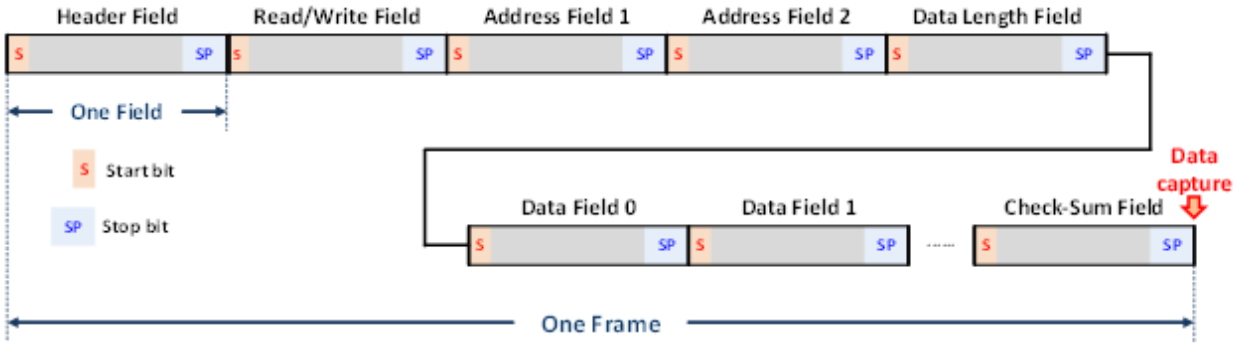

Figure 5. Frame Format for Write Mode

In read mode, the Frame format is different from write mode in terms of the gap. The following 2 types of gaps are inserted in this mode as shown in Figure 6.

- Gap1; Between "Data Length Field" and "Data Field 0"
- Gap2; Between "Check−sum Field" and "Data Field N" (N = 8, 16, 24, 32, 40, 48 or 56)

The Gap1 length is total 11.5 bits precisely as shown in Figure 6, add 0.5 bit after stop bit, though Gap2 length is total 11bits. The communication pin keeps high during these gaps.

Figure 6. Frame Format for Read Mode

Once the slave enters “Communication“ mode, an equivalent time of 3 Fields is allowable as a delay between the previously transferred Field and current transferring Field in both read and write mode. If the delay is longer than 3 Fields, “Time Out Error“ is flagged, then the slave goes to “Standby“ mode.

Table 1 is the description of each Field.

Table 1. FIELD DESCRIPTION

Field Name	Function	Comment
Header	Start SWAN communication	The data pattern is 0x40.
R/W	Set Read / Write mode	Select the Read / Write mode
Address	Set target address	Need 2 Fields because of 2 bytes address
Data Length	Set data length	Set the data length. Max length is 64 bytes.
Data	Set data value	1 byte per 1 Field
Check−Sum	Set check−sum value	Complete the communication and the transferred data is written into the target register. If the data length is longer than 8 bytes, the Check−sum should be added every 8 bytes data and remained bytes.

In read mode, the Master must send from “Header Field“ to “Data Length Field“ to the Slave, then the Slave sends the “Data Field“ and “Check−Sum Field“ to the Master. On the other hand, the Master transfers from Header Field to Check−Sum Field in write mode. The amount of Data Field depends on the data length specified by Data Length Field. Due to the Frame format configuration, it is obviously impossible to operate register read and write at the same time.

DESCRIPTION OF THE “FIELD”

The followings are the explanation of the each “Field“ configures the “Frame“.

Header Field

Figure 7 shows the configuration of “Header Field“.

Figure 7. Header Field

The data (D7 to D0) is 0x40 (= 0100 0000). This Field activates the SWAN at the device Power ON or at the state transition from “Standby“ to “Communication“ mode.

At the Slave power on, the Master inputs this Field repeatedly at least minimum times as eq.1 after 1ms from power on to make the Slave fix the baud rate. After fixing the baud rate, the Slave goes to “Power−on Standby“ mode and waits for the next “R/W Field“.

At the transition from “Standby” to “Communication” mode, it is no need to input the repetition of this Header Field, just 1 Header Field is needed and the next R/W Field is input sequentially.

R/W Field

“R/W Field” defines read/write mode. Figure 8 is the configuration of “R/W Field“. To set 0 to D0 means read mode and 1 means write mode. D [5:1] should be 0. P0 in D6 and P1 in D7 are the parity bits calculated by eq.2 and eq.3. Consequently, R/W Field has only 2 patterns as Figure 9.

$$P0 = D0 \text{ xor } D1 \text{ xor } D2 \text{ xor } D4 \quad \text{(eq. 2)}$$

$$P1 = \neg(D1 \text{ xor } D3 \text{ xor } D4 \text{ xor } D5) \quad \text{(eq. 3)}$$

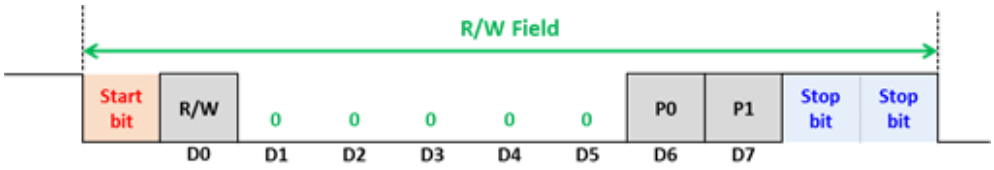

Figure 8. R/W Field

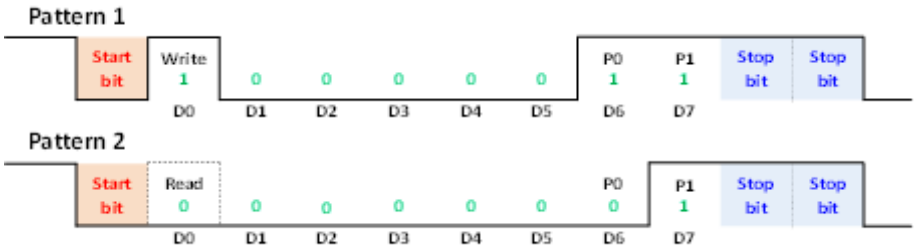

Figure 9. R/W Field pattern

When the Slave receives this Field after detecting Header Field at power on, it moves to “Communication“ mode where the communication starts. In this case, the delay represented by eq.4 is allowed exceptionally between Header Field and R/W Field, which means the delay equivalent to 9% of 1 Field is accepted.

$$Inv < 0.09 \times Ft \quad \text{(eq. 4)}$$

where
Inv = Interval between the Fields [s]
Ft = Input Field length [s]

On the other hand, in transition from “Standby“ to “Communication“ mode, no gap is allowed between Header Field and R/W Field.

Once the Slave moves to “Communication“ mode, it allows the time corresponding to 3 Fields as the maximum delay between each Field. During this delay, please keep the communication pin high to prevent the communication error.

Address Field

"Address Field" has 2 Fields; "Address Field 1" and "Address Field 2" because the register address is 2 bytes. The lower 8 bits of the data address is placed in the D [7:0] in the Address Field 1 and the upper 8bits is placed in the D [7:0] in the Address Field 2. Figure 10 shows the configuration of the Address Field.

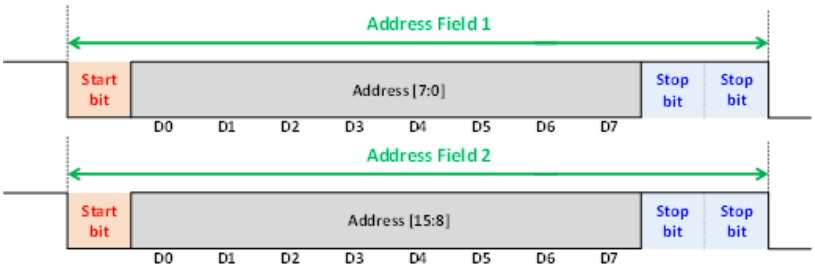

Figure 10. Address field format

Data Length Field

It defines the data length of the transferred data. The maximum length is 64bytes and D [5:0] defines it as below. D [7:6] are parity bits, P0 and P1 calculated by eq.2 and eq. 3. Figure 11 shows the configuration of the "Data Length Field".

- D[5:0] 000000 → 1 byte
- D[5:0] 111111 → 64 bytes

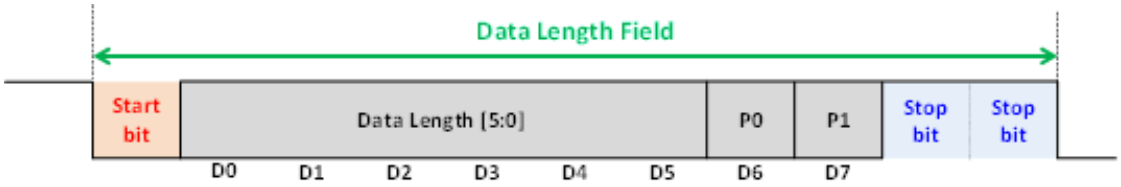

Figure 11. Data length field format

The register address specified by the two Address Fields is the first address for the multiple bytes transformation. The address must be one incremental consecutively for all bytes specified by this Data Length Field.

Data Field

The data is transferred by each 1byte. In read mode, the data is transferred from the target register specified by the "Address Field" to the Master. In write mode, the data is transferred to the target register of the Slave. If the data length specified by the "Data Length Field" is longer than 1 byte, the multi−bytes corresponding to the data amount are transferred sequentially. Figure 12 shows the configuration of the Data Field.

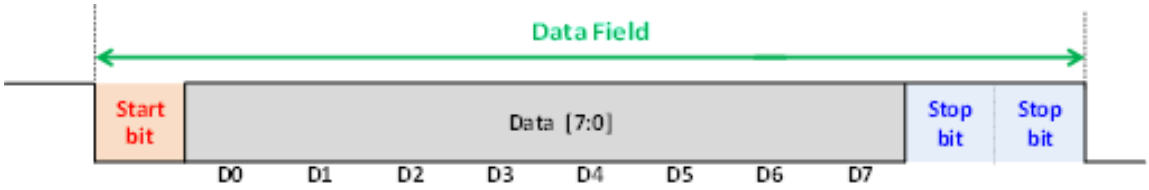

Figure 12. Data field format

Check−Sum Field

If the data length is longer than 8 bytes, the Check−sum should be added every 8 bytes and remained bytes. "Check−Sum Field" is transferred after "Data Field". The value is the inversion of the summation of the D [7:0] for all the Field except for "Header Field". If the summation has an overflow of the most significant digit of the byte, it is added to the least significant digit. In write mode, the Slave receives the check−sum value from the Master and compares it. When it is correct, the register write is happened. If it is not correct, the "Check−Sum Error" is flagged and transferred data is discarded. In read mode, the Slave sends the check−sum value to the Master and it is useful to check the read data validity. Figure 13 shows the configuration of the Check−Sum Field.

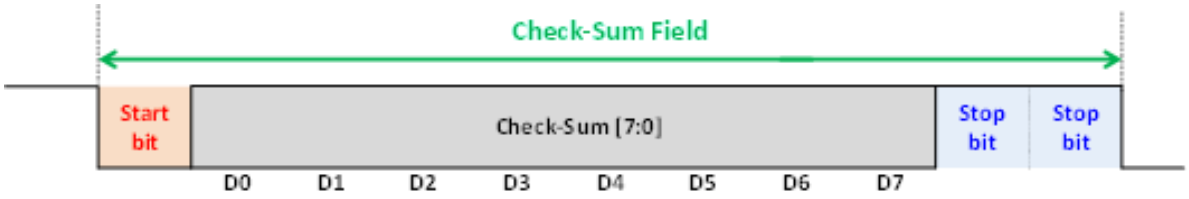

Figure 13. Check−Sum field format

The check−sum value is calculated by each 8 Data Fields. Figure 14 shows the example of the appearance of the check−sum in various cases.

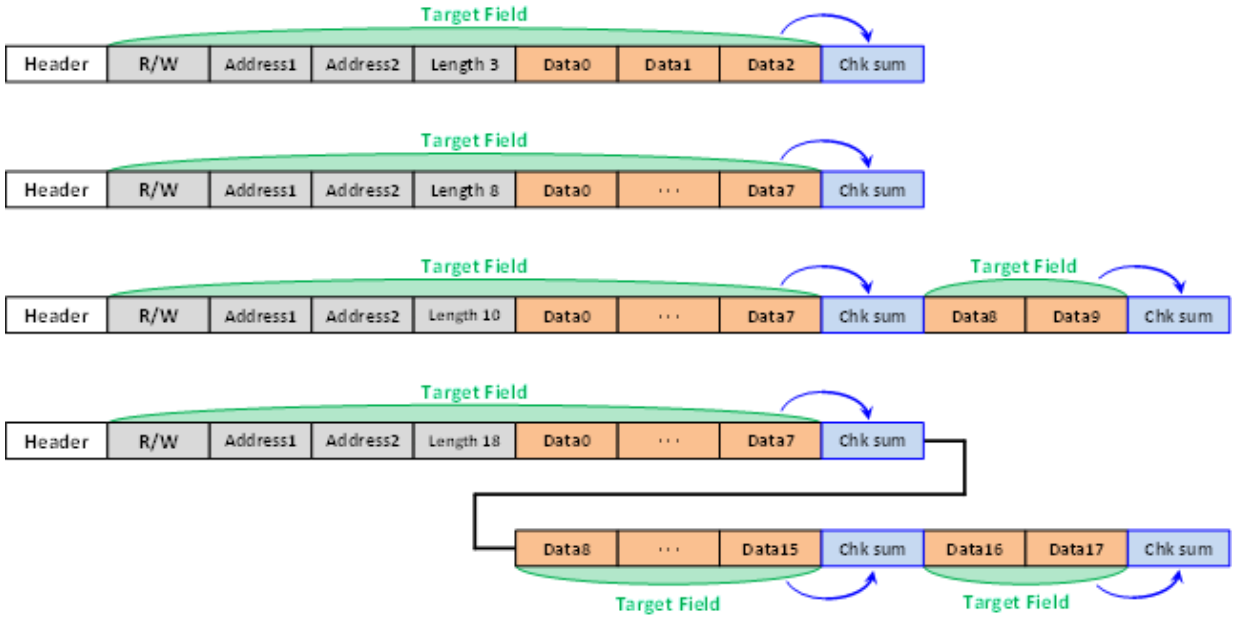

Figure 14. The relationship between the number of field and the Check−Sum field

From Figure 14, the maximum target Fields for the first check−sum value are 12 Fields, from R/W Field to Data Field 7. After that, the check sum calculation is done by each 8 Data Fields. Every time the check−sum is finished successfully, the register write is happened. If the check−sum value is not correct, the “Check−Sum Error” is flagged. Figure 15 shows the example of check−sum error occurring in write mode. In this case, the Master transfers 3 check−sum values. 1st value is correct then the 8 bytes data corresponding to Data Field 0 to 7 are written into the register specified by Address Field 1 and 2 of the Slave. However, 2nd value is not correct, then the data supposed to be transferred to Data Field 8 to 15 is discarded and Data Field 16 and Data Field 17 are also not transferred to the Slave because communication has already terminated by the error. In this case, the registers where the values of Data Field 8 to 17 are supposed to be written keep previous values.

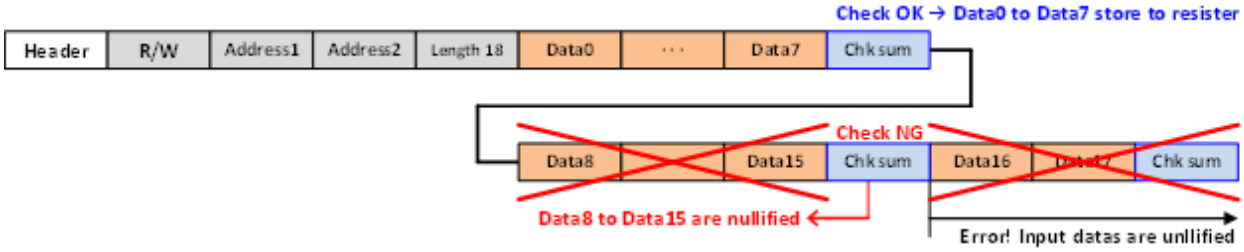

Figure 15. Example when the Check−sum error occurred

Following is a specific example of the Frame of write mode. Target address and the data is:

- Reg. 0x1005, Data 1011 1001
- Reg. 0x1006, Data 0010 1100

In this case, Header Field and R/W Field is uniquely determined as shown in Figure 16.

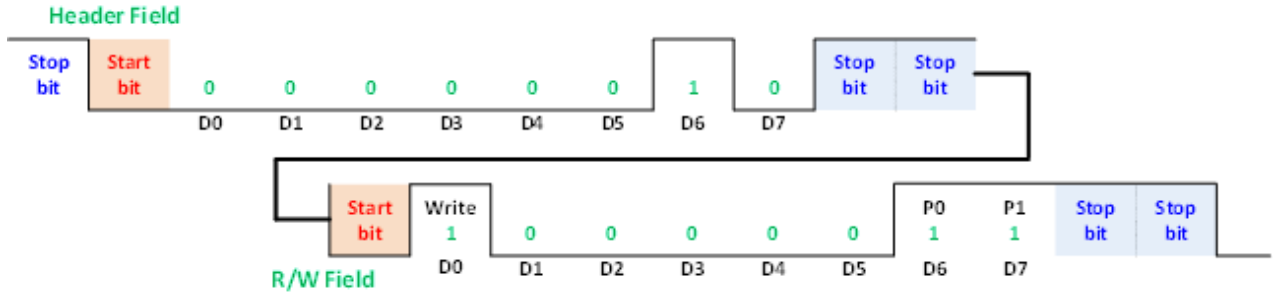

Figure 16. Specific example of header field and R/W field

For the Address Field, lower 8 bits are 0x05 and upper 8bits are 0x10, then Address Field 1 and Address Field 2 are set as shown in Figure 17.

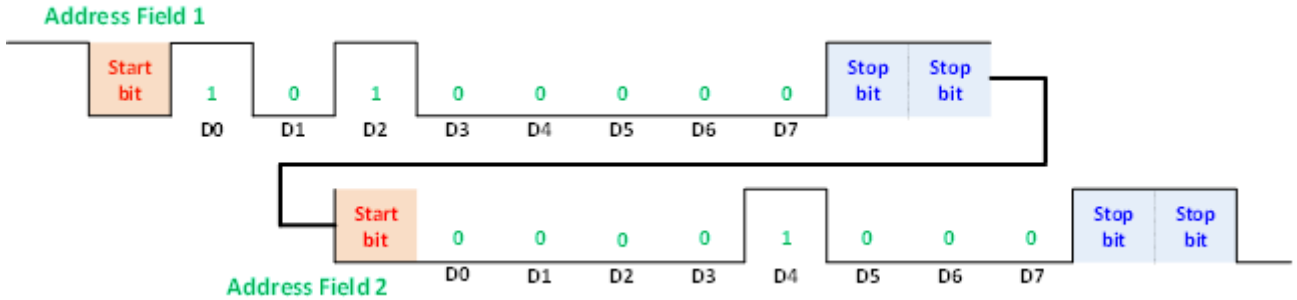

Figure 17. Specific example of address fields

The data length is 2 bytes, then D [5:0] of Data Length Field is set 000001 as shown in Figure 18. The parity P0 in D[6] is (1 xor 0 xor 0 xor 0) = 1, P1 in D[7] is (0 xor 0 xor 0 xor 0) = 1.

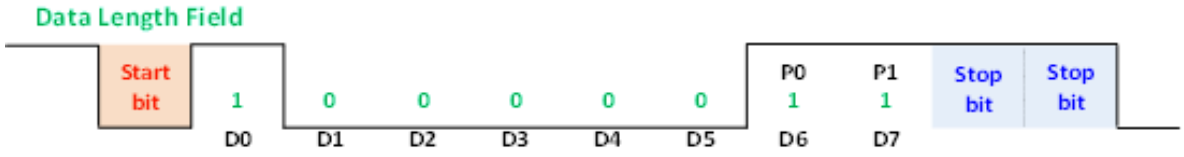

Figure 18. Specific example of address fields

Data Field is set as shown in Figure 19.

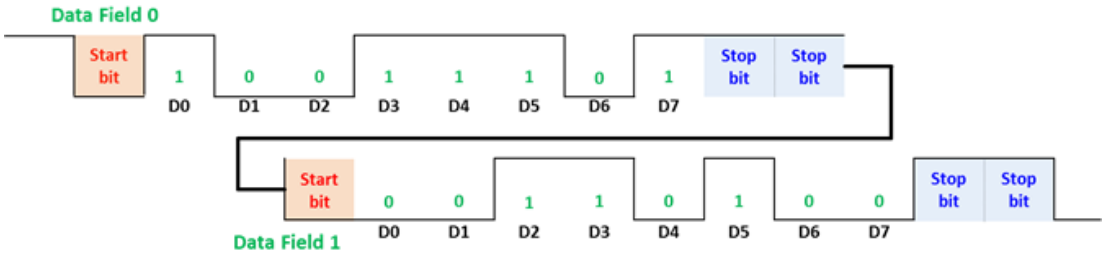

Figure 19. Specific example of data fields

All Fields are shown in Figure 20. The check−sum value is to invert the summation of all D [7:0] in each Field except for Header Field. Figure 21 shows the check−sum calculation.

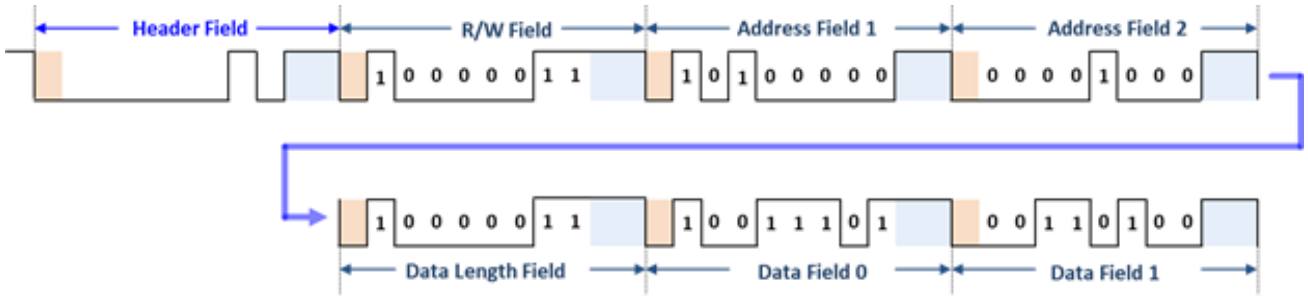

Figure 20. All Fields from the header field to Data Field 1

	R/W Field	1	1	0	0	0	0	0	1
+)	Address Field 1	0	0	0	0	0	1	0	1
	Sum 1	1	1	0	0	0	1	1	0
	Sum 1	1	1	0	0	0	1	1	0
+)	Address Field 2	0	0	0	1	0	0	0	0
	Sum 2	1	1	0	1	0	1	1	0
	Sum 2	1	1	0	1	0	1	1	0
+)	Data Length Field	1	1	0	0	0	0	0	1
	Sum 3 1	1	0	0	1	0	1	1	1
	Sum 3	1	0	0	1	1	0	0	0
+)	Data Field 0	1	0	1	1	1	0	0	1
	Sum 4 1	0	1	0	1	0	0	0	1
	Sum 4	0	1	0	1	0	0	1	0
+)	Data Field 1	0	0	1	0	1	1	0	0
	Sum 5	0	1	1	1	1	1	1	0
	$\overline{\text{Sum 5}}$	1	0	0	0	0	0	0	1
	Check−Sum Data	1	0	0	0	0	0	0	1

Figure 21. Calculation for check−sum

Sum5 in Figure 21 shows the summation of the all D [7:0], then check−sum value is 1000 0001. From the result, the "Check−sum Field" is as shown in Figure 22. Finally, the data Frame is made by adding Figure 22 to Figure 20.

Figure 22. Specific example of check−sum fields

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910

ON Semiconductor Website: **www.onsemi.com**

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative