

## 使用结构化数据类型时的填充字节

问题：当进行串口通讯指令或开放式用户通讯指令接收数据时，如果将用户自定义数据类型的变量用作接收数据区时，为什么接收数据长度是正确的，但是接收到的数据反而会丢失部分字节？

以两个 S7-1500 的 ISO-ON-TCP 为例，两侧分别调用 TSEND\_C 和 TRCV\_C，如下图所示，发送侧发送的数据如下：

Data_block_2				
	Name	Data type	Start value	Monitor value
1	Static			
2	test1	Array[0..41] of Byte		
3	test1[0]	Byte	1	16#01
4	test1[1]	Byte	2	16#02
5	test1[2]	Byte	3	16#03
6	test1[3]	Byte	4	16#04
7	test1[4]	Byte	5	16#05
8	test1[5]	Byte	6	16#06
9	test1[6]	Byte	7	16#07
10	test1[7]	Byte	8	16#08
11	test1[8]	Byte	9	16#09
12	test1[9]	Byte	0	16#00
13	test1[10]	Byte	1	16#01
14	test1[11]	Byte	2	16#02
15	test1[12]	Byte	3	16#03
16	test1[13]	Byte	4	16#04
17	test1[14]	Byte	5	16#05
18	test1[15]	Byte	6	16#06
19	test1[16]	Byte	7	16#07
20	test1[17]	Byte	8	16#08
21	test1[18]	Byte	9	16#09
22	test1[19]	Byte	0	16#00
23	test1[20]	Byte	1	16#01
24	test1[21]	Byte	2	16#02
25	test1[22]	Byte	3	16#03
26	test1[23]	Byte	4	16#04
27	test1[24]	Byte	5	16#05
28	test1[25]	Byte	6	16#06
29	test1[26]	Byte	7	16#07
30	test1[27]	Byte	8	16#08
31	test1[28]	Byte	9	16#09

接收数据区的用户定义数据类型变量及结构如下：

Data_block_1				
	Name	Data type	Start value	Monitor value
1	Static			
2	test	"UDT_1"		
3	struct1	Struct		
4	array1	Array[0..5] of Byte		
5	struct2	Struct		
6	array1	Array[0..6] of Byte		
7	struct3	Struct		
8	array1	Array[0..6] of Byte		
9	struct4	Struct		
10	array1	Array[0..5] of Byte		
11	struct5	Struct		
12	array1	Array[0..5] of Byte		
13	struct6	Struct		
14	array1	Array[0..6] of Byte		

接收側接收到的数据如下:

Data_block_1									
	Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	test	"UDT_1"			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	struct1	Struct			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	array1	Array[0..5] of Byte			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	array1[0]	Byte	16#0	16#01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	array1[1]	Byte	16#0	16#02	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	array1[2]	Byte	16#0	16#03	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	array1[3]	Byte	16#0	16#04	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	array1[4]	Byte	16#0	16#05	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	array1[5]	Byte	16#0	16#06	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	struct2	Struct			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	array1	Array[0..6] of Byte			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	array1[0]	Byte	16#0	16#07	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	array1[1]	Byte	16#0	16#08	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	array1[2]	Byte	16#0	16#09	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	array1[3]	Byte	16#0	16#00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	array1[4]	Byte	16#0	16#01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	array1[5]	Byte	16#0	16#02	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	array1[6]	Byte	16#0	16#03	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	struct3	Struct			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	array1	Array[0..6] of Byte			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22	array1[0]	Byte	16#0	16#05	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	array1[1]	Byte	16#0	16#06	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	array1[2]	Byte	16#0	16#07	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25	array1[3]	Byte	16#0	16#08	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26	array1[4]	Byte	16#0	16#09	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
27	array1[5]	Byte	16#0	16#00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
28	array1[6]	Byte	16#0	16#01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
29	struct4	Struct			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
30	array1	Array[0..5] of Byte			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
31	array1[0]	Byte	16#0	16#03	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
32	array1[1]	Byte	16#0	16#04	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
33	array1[2]	Byte	16#0	16#05	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
34	array1[3]	Byte	16#0	16#06	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
35	array1[4]	Byte	16#0	16#07	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

如果要在 SIMATIC CPU 和外部设备之间传输预定义形式和大小结构化数据类型，则可以在内部通过默认存储区完成。

使用 SIMATIC，所有结构化数据类型都将与偶数起始地址对齐（可被 2 整除）。因此，结构化数据类型 BYTE 中选定的长度也应被 2 整除。

如果由于先前的 BYTE 或结构化数据类型的长度的原因，结构化数据类型的起始地址转换为奇数地址或结束地址转换为偶数地址，则填充字节将自动插入到相关位置。

**提示**

填充字节对于数据存储不可见（例如在 DB 中），仅影响地址分配（在默认存储区中）。

举例来说，如果长度为七个字节的数组在数据块中跟随三个单字节，则会在数组前后均自动插入一个填充字节。例如，前三个字节位于地址“0.0”到“2.0”。而后面的数组位于地址“4.0”到“10.0”。随后的单个字节将获得地址“12.0”。

**结构化数据类型填充字节的概述**

在 TIA Portal 中，以下组件受填充字节的影响：

主题	数据类型	CPU	数据存储	指令示例	示例场景
填充字节	Struct、PLC 数据类型、数组	S7-300、 S7-400、 S7-1200、 S7-1500	非优化组件、优化组件	Serialize、 Deserialize、 WRREC、RDREC、 TSEND_C、TRCV_C、PUT、 GET	<ul style="list-style-type: none"> <li>• 具有结构化元素的结构</li> <li>• I/O（制造商特定数据记录）</li> <li>• I/O 设备地址空间中的结构</li> <li>• 与其它制造商的 I/O 进行通信</li> <li>• 在 Linux 下使用硬件设备的驱动程序</li> </ul>

**问题情况示例**

已创建以下嵌套结构作为“typeHARTrequest”PLC 数据类型，以作为数据记录发送到外部设备。（见绿色边框。）

HART_DATA					
	Name	Datentyp	Offset	Startwert	Beobachtungswert
1	Static				
2	statCommand	Array[0..10] of Byte	0.0		
3	statCommand[0]	Byte	0.0	16#0	16#00
4	statCommand[1]	Byte	1.0	BYTE#16#05	16#05
5	statCommand[2]	Byte	2.0	BYTE#16#82	16#82
6	statCommand[3]	Byte	3.0	BYTE#16#AA	16#AA
7	statCommand[4]	Byte	4.0	BYTE#16#13	16#13
8	statCommand[5]	Byte	5.0	BYTE#16#AA	16#AA
9	statCommand[6]	Byte	6.0	BYTE#16#85	16#85
10	statCommand[7]	Byte	7.0	BYTE#16#3F	16#3F
11	statCommand[8]	Byte	8.0	BYTE#16#03	16#03
12	statCommand[9]	Byte	9.0	16#0	16#00
13	statCommand[10]	Byte	10.0	BYTE#16#28	16#28
14	statHARTreply	Array[0..239] of Byte	12.0		
15	ComendUDT	"typeHARTrequest"	252.0		
16	req_control	Byte	252.0	16#0	16#00
17	preamble_bytes	Byte	253.0	16#05	16#05
18	start_character	Byte	254.0	16#82	16#82
19	address	Array[0..4] of Byte	256.0		
20	address[0]	Byte	256.0	BYTE#16#AA	16#AA
21	address[1]	Byte	257.0	BYTE#16#13	16#13
22	address[2]	Byte	258.0	BYTE#16#AA	16#AA
23	address[3]	Byte	259.0	BYTE#16#85	16#85
24	address[4]	Byte	260.0	BYTE#16#3F	16#3F
25	command	Byte	262.0	BYTE#16#03	16#03
26	length	Byte	263.0	16#0	16#00
27	check_sum	Byte	264.0	BYTE#16#28	16#28
28	<Hinzugefügt>				

Length: 11 Byte + 2 Byte + 1 Byte

Offset 255.0  
Padding Byte

Offset 261.0  
Padding Byte

问题：“typeHARTrequest”PLC 数据类型的实际长度为 11 个字节。以字节为单位的几个奇数长度也会导致在 PLC 数据类型中填充字节。（见红色标记。）包括所有填充字节在内，PLC 数据类型有 14 个字节的长度用于数据交换。因此，SIMATIC CPU 的寻址地址范围将与外部设备用于数据交换的地址范围不匹配。

解决方法：将标记为蓝色的数组更改为五个单字节。然后，新“typeHARTrequest”PLC 数据类型的长度为 12 个字节（包括末尾的填充字节）。