

Totally Integrated Automation Portal								
AKD_PNU_v02 [FB2]								
AKD_PNU_v02 Properties								
General								
Name	AKD_PNU_v02	Number	2	Type	FB	Language	SCL	
Numbering	automatic							
Information								
Title	AKD_PNU_v02	Author	jcoleman	Comment	This function block uses the Parameter Access channel to read and write PNU parameters values.	Family	PNU	
Version	2.0	User-defined ID						
Name	Data type	Offset	Default value	Accessible from HMI	Visible in HMI	Setpoint	Comment	
▼ Input								
iID	HW_IO	0.0	16#0	False	False	False	unique connection ID	
iStartRequest	Bool	2.0	false	False	False	False	start Read or Write	
iRequestType	Bool	2.1	false	False	False	False	0 = Read; 1 = Write	
iFormat	Int	4.0	0	False	False	False	Datatype 0x41 = Byte, 0x42 = Word, 0x43 = DWord	
iPNUNumber	Word	6.0	16#0	False	False	False	PNU address	
iPNUsubIndex	Word	8.0	16#0	False	False	False	PNU Subindex	
▼ Output								
oActive	Bool	10.0	false	False	False	False	request in progress	
oDone	Bool	10.1	false	False	False	False	request finished	
oError	Bool	10.2	false	False	False	False	request failed	
oStatus	DWord	12.0	16#0	False	False	False	status of the failed function	
▼ InOut								
ioData	DWord	16.0	16#0	False	False	False		
▼ Static								
▼ PNURead	RDREC			False	False	False		
▼ Input								
REQ	Bool		FALSE	False	False	False	REQ = 1: Transfer data record	
ID	HW_IO		16#0	False	False	False	HW-Id of the DP slave/PROFINET IO component	
INDEX	DInt		0	False	False	False	Data record number	
MLEN	UInt		0	False	False	False	Maximum length in bytes of the data	
▼ Output								
VALID	Bool		FALSE	False	False	False	Function performed	
BUSY	Bool		FALSE	False	False	False	Function busy	
ERROR	Bool		FALSE	False	False	False	Error flag	
STATUS	DWord		DW#16#0	False	False	False	Function result/error message	
LEN	UInt		0	False	False	False	Length of the fetched data record	
▼ InOut								
RECORD	Variant			False	False	False	Target area for the fetched data record	
Static								
▼ PNUWrite	WRREC			False	False	False		
▼ Input								
REQ	Bool		FALSE	False	False	False	REQ = 1: Transfer data record	
ID	HW_IO		16#0	False	False	False	HW-Id of the DP slave/PROFINET IO component	
INDEX	DInt		0	False	False	False	Data record number	
LEN	UInt		0	False	False	False	Maximum length in bytes of the data	
▼ Output								
DONE	Bool		FALSE	False	False	False	Function performed	
BUSY	Bool		FALSE	False	False	False	Function busy	
ERROR	Bool		FALSE	False	False	False	Error flag	
STATUS	DWord		DW#16#0	False	False	False	Function result/error message	
▼ InOut								
RECORD	Variant			False	False	False	Data record	
Static								
bDoWrite	Bool	20.0	false	False	False	False		
bDoRead	Bool	20.1	false	False	False	False		
▼ byReadArray	Array[0..147] of Byte	22.0		False	False	False		
byReadArray[0]	Byte	0.0	16#0	False	False	False		
byReadArray[1]	Byte	1.0	16#0	False	False	False		
byReadArray[2]	Byte	2.0	16#0	False	False	False		
byReadArray[3]	Byte	3.0	16#0	False	False	False		
byReadArray[4]	Byte	4.0	16#0	False	False	False		
byReadArray[5]	Byte	5.0	16#0	False	False	False		
byReadArray[6]	Byte	6.0	16#0	False	False	False		
byReadArray[7]	Byte	7.0	16#0	False	False	False		
byReadArray[8]	Byte	8.0	16#0	False	False	False		

Totally Integrated Automation Portal								
Name		Data type	Offset	Default value	Accessible from HMI	Visible in HMI	Setpoint	Comment
byReadArray[87]		Byte	87.0	16#0	False	False	False	
byReadArray[88]		Byte	88.0	16#0	False	False	False	
byReadArray[89]		Byte	89.0	16#0	False	False	False	
byReadArray[90]		Byte	90.0	16#0	False	False	False	
byReadArray[91]		Byte	91.0	16#0	False	False	False	
byReadArray[92]		Byte	92.0	16#0	False	False	False	
byReadArray[93]		Byte	93.0	16#0	False	False	False	
byReadArray[94]		Byte	94.0	16#0	False	False	False	
byReadArray[95]		Byte	95.0	16#0	False	False	False	
byReadArray[96]		Byte	96.0	16#0	False	False	False	
byReadArray[97]		Byte	97.0	16#0	False	False	False	
byReadArray[98]		Byte	98.0	16#0	False	False	False	
byReadArray[99]		Byte	99.0	16#0	False	False	False	
byReadArray[100]		Byte	100.0	16#0	False	False	False	
byReadArray[101]		Byte	101.0	16#0	False	False	False	
byReadArray[102]		Byte	102.0	16#0	False	False	False	
byReadArray[103]		Byte	103.0	16#0	False	False	False	
byReadArray[104]		Byte	104.0	16#0	False	False	False	
byReadArray[105]		Byte	105.0	16#0	False	False	False	
byReadArray[106]		Byte	106.0	16#0	False	False	False	
byReadArray[107]		Byte	107.0	16#0	False	False	False	
byReadArray[108]		Byte	108.0	16#0	False	False	False	
byReadArray[109]		Byte	109.0	16#0	False	False	False	
byReadArray[110]		Byte	110.0	16#0	False	False	False	
byReadArray[111]		Byte	111.0	16#0	False	False	False	
byReadArray[112]		Byte	112.0	16#0	False	False	False	
byReadArray[113]		Byte	113.0	16#0	False	False	False	
byReadArray[114]		Byte	114.0	16#0	False	False	False	
byReadArray[115]		Byte	115.0	16#0	False	False	False	
byReadArray[116]		Byte	116.0	16#0	False	False	False	
byReadArray[117]		Byte	117.0	16#0	False	False	False	
byReadArray[118]		Byte	118.0	16#0	False	False	False	
byReadArray[119]		Byte	119.0	16#0	False	False	False	
byReadArray[120]		Byte	120.0	16#0	False	False	False	
byReadArray[121]		Byte	121.0	16#0	False	False	False	
byReadArray[122]		Byte	122.0	16#0	False	False	False	
byReadArray[123]		Byte	123.0	16#0	False	False	False	
byReadArray[124]		Byte	124.0	16#0	False	False	False	
byReadArray[125]		Byte	125.0	16#0	False	False	False	
byReadArray[126]		Byte	126.0	16#0	False	False	False	
byReadArray[127]		Byte	127.0	16#0	False	False	False	
byReadArray[128]		Byte	128.0	16#0	False	False	False	
byReadArray[129]		Byte	129.0	16#0	False	False	False	
byReadArray[130]		Byte	130.0	16#0	False	False	False	
byReadArray[131]		Byte	131.0	16#0	False	False	False	
byReadArray[132]		Byte	132.0	16#0	False	False	False	
byReadArray[133]		Byte	133.0	16#0	False	False	False	
byReadArray[134]		Byte	134.0	16#0	False	False	False	
byReadArray[135]		Byte	135.0	16#0	False	False	False	
byReadArray[136]		Byte	136.0	16#0	False	False	False	
byReadArray[137]		Byte	137.0	16#0	False	False	False	
byReadArray[138]		Byte	138.0	16#0	False	False	False	
byReadArray[139]		Byte	139.0	16#0	False	False	False	
byReadArray[140]		Byte	140.0	16#0	False	False	False	
byReadArray[141]		Byte	141.0	16#0	False	False	False	
byReadArray[142]		Byte	142.0	16#0	False	False	False	
byReadArray[143]		Byte	143.0	16#0	False	False	False	
byReadArray[144]		Byte	144.0	16#0	False	False	False	
byReadArray[145]		Byte	145.0	16#0	False	False	False	
byReadArray[146]		Byte	146.0	16#0	False	False	False	
byReadArray[147]		Byte	147.0	16#0	False	False	False	
▼ byWriteArray		Array[0..31] of Byte	170.0		False	False	False	
byWriteArray[0]		Byte	0.0	16#0	False	False	False	
byWriteArray[1]		Byte	1.0	16#0	False	False	False	
byWriteArray[2]		Byte	2.0	16#0	False	False	False	
byWriteArray[3]		Byte	3.0	16#0	False	False	False	
byWriteArray[4]		Byte	4.0	16#0	False	False	False	
byWriteArray[5]		Byte	5.0	16#0	False	False	False	
byWriteArray[6]		Byte	6.0	16#0	False	False	False	
byWriteArray[7]		Byte	7.0	16#0	False	False	False	
byWriteArray[8]		Byte	8.0	16#0	False	False	False	
byWriteArray[9]		Byte	9.0	16#0	False	False	False	
byWriteArray[10]		Byte	10.0	16#0	False	False	False	
byWriteArray[11]		Byte	11.0	16#0	False	False	False	
byWriteArray[12]		Byte	12.0	16#0	False	False	False	
byWriteArray[13]		Byte	13.0	16#0	False	False	False	
byWriteArray[14]		Byte	14.0	16#0	False	False	False	

Totally Integrated Automation Portal

Name	Data type	Offset	Default value	Accessible from HMI	Visible in HMI	Setpoint	Comment
byWriteArray[15]	Byte	15.0	16#0	False	False	False	
byWriteArray[16]	Byte	16.0	16#0	False	False	False	
byWriteArray[17]	Byte	17.0	16#0	False	False	False	
byWriteArray[18]	Byte	18.0	16#0	False	False	False	
byWriteArray[19]	Byte	19.0	16#0	False	False	False	
byWriteArray[20]	Byte	20.0	16#0	False	False	False	
byWriteArray[21]	Byte	21.0	16#0	False	False	False	
byWriteArray[22]	Byte	22.0	16#0	False	False	False	
byWriteArray[23]	Byte	23.0	16#0	False	False	False	
byWriteArray[24]	Byte	24.0	16#0	False	False	False	
byWriteArray[25]	Byte	25.0	16#0	False	False	False	
byWriteArray[26]	Byte	26.0	16#0	False	False	False	
byWriteArray[27]	Byte	27.0	16#0	False	False	False	
byWriteArray[28]	Byte	28.0	16#0	False	False	False	
byWriteArray[29]	Byte	29.0	16#0	False	False	False	
byWriteArray[30]	Byte	30.0	16#0	False	False	False	
byWriteArray[31]	Byte	31.0	16#0	False	False	False	
bPNUWriteBusy	Bool	202.0	false	False	False	False	
bPNUWriteDone	Bool	202.1	false	False	False	False	
bPNUWriteError	Bool	202.2	false	False	False	False	
dwPNUWriteStatus	DWord	204.0	16#0	False	False	False	
bPNUReadBusy	Bool	208.0	false	False	False	False	
bPNUReadDone	Bool	208.1	false	False	False	False	
bPNUReadError	Bool	208.2	false	False	False	False	
dwPNUReadStatus	DWord	210.0	16#0	False	False	False	
wDataLenght	Word	214.0	16#0	False	False	False	
bReqActive	Bool	216.0	false	False	False	False	
wLenght	Word	218.0	16#0	False	False	False	
byReference	Byte	220.0	16#0	False	False	False	
bReadActive	Bool	221.0	false	False	False	False	
bWriteActive	Bool	221.1	false	False	False	False	
Temp							
Constant							

```
0001 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
0002 // Communication module between a Kollmorgen AKD and S7-1200/1500 PLC, for direct parameter access  
0003 //  
0004 // Version When Who What  
0005 // -----  
0006 // V 2.000 17.07.06 JColeman KM Changed input and output names  
0007 // V 1.000 17.09.15 MRupf KM Conversion to TIA V13 SP1  
0008 // V 0.005 14.07.15 MRupf KM Automatic format check for read values  
0009 // V 0.004 06.07.15 MRupf KM oDone and oActive Trigger corrected  
0010 // V 0.002 26.02.15 MRupf KM iSubIndex added  
0011 // V 0.001 18.02.15 MRupf KM first version  
0012 ///////////////////////////////////////////////////////////////////////  
0013  
0014 // Set the output in error case and only reset with new req  
0015 IF #bPNUReadError OR #bPNUWriteError OR (#iFormat - 16#41 < 0) OR (#iFormat - 16#43 > 0) THEN  
0016     #oError := TRUE;  
0017     IF #bPNUReadError THEN  
0018         #bDoRead := FALSE;  
0019         #oStatus := #dwPNUReadStatus;  
0020     END_IF;  
0021     IF #bPNUWriteError THEN  
0022         #bDoWrite := FALSE;  
0023         #oStatus := #dwPNUWriteStatus;  
0024     END_IF;  
0025 END_IF;  
0026  
0027 // Request edge starts a new call if none active  
0028 IF #iStartRequest AND NOT #bReqActive THEN  
0029     #byReference := #byReference + 1;      // Every request gets a new reference for debugging purpose  
0030     IF #byReference = 0 THEN  
0031         #byReference := #byReference + 1;  
0032     END_IF;  
0033  
0034     // Reset all the status bit with every request  
0035     #oActive := TRUE;  
0036     #oDone := FALSE;  
0037     #oError := FALSE;  
0038     #oStatus := 0;  
0039  
0040     FILL_BLK(IN := 0,  
0041             COUNT := 16,  
0042             OUT => #byWriteArray[0]);  
0043  
0044     FILL_BLK(IN := 0,  
0045             COUNT := 148,  
0046             OUT => #byReadArray[0]);  
0047  
0048     // Set the data to start a PNU write request  
0049     IF #iRequestType THEN
```

Totally Integrated Automation Portal		
<pre>0050 #byWriteArray[0] := #byReference; // Byte 0 reference 0051 #byWriteArray[1] := 16#02; // Byte 1 Request ID (2 = write) 0052 #byWriteArray[2] := 16#00; // Byte 2 Axis (0) 0053 #byWriteArray[3] := 16#01; // Byte 3 Number of parameters 0054 #byWriteArray[4] := 16#10; // Byte 4 Attribute 0055 #byWriteArray[5] := 16#01; // Byte 5 Number of elements 0056 #byWriteArray[6] := #iPNUnumber.%B1; // Byte 6&7 PNU number 0057 #byWriteArray[7] := #iPNUnumber.%B0; // Byte 6&7 PNU number 0058 #byWriteArray[8] := #iPNUsubIndex.%B1; // Byte 8&9 Subindex (0) 0059 #byWriteArray[9] := #iPNUsubIndex.%B0; // Byte 8&9 Subindex (0) 0060 #byWriteArray[10] := #iFormat; // Byte 10 Format 0061 #byWriteArray[11] := 16#01; // Byte 11 Number of values 0062 0063 // Set the lenght automatically according to the Format 0064 IF #iFormat = 16#41 THEN 0065 #wLenght := 16#0E; 0066 #byWriteArray[12] := #ioData.%B0; // Byte 12 Value 0067 ELSE 0068 IF #iFormat = 16#42 THEN 0069 #wLenght := 16#10; 0070 #byWriteArray[12] := #ioData.%B1; // Byte 12&13 Value 0071 #byWriteArray[13] := #ioData.%B0; // Byte 12&13 Value 0072 ELSE 0073 IF #iFormat = 16#43 THEN 0074 #wLenght := 16#14; 0075 #byWriteArray[12] := #ioData.%B3; // Byte 12&13 Value 0076 #byWriteArray[13] := #ioData.%B2; // Byte 12&13 Value 0077 #byWriteArray[14] := #ioData.%B1; // Byte 14&15 Value 0078 #byWriteArray[15] := #ioData.%B0; // Byte 14&15 Value 0079 END_IF; 0080 END_IF; 0081 END_IF; 0082 0083 #bDoWrite := TRUE; 0084 0085 // Set the data to start a PNU read request 0086 ELSE 0087 #byWriteArray[0] := #byReference; // Byte 0 reference 0088 #byWriteArray[1] := 16#01; // Byte 1 Request ID (1 = read) 0089 #byWriteArray[2] := 16#00; // Byte 2 Axis (0) 0090 #byWriteArray[3] := 16#01; // Byte 3 Number of parameters 0091 #byWriteArray[4] := 16#10; // Byte 4 Attribute 0092 #byWriteArray[5] := 16#01; // Byte 5 Number of elements 0093 #byWriteArray[6] := #iPNUnumber.%B1; // Byte 6&7 PNU number 0094 #byWriteArray[7] := #iPNUnumber.%B0; // Byte 6&7 PNU number 0095 #byWriteArray[8] := #iPNUsubIndex.%B1; // Byte 8&9 Subindex (0) 0096 #byWriteArray[9] := #iPNUsubIndex.%B0; // Byte 8&9 Subindex (0) 0097 0098 #ioData := 0; 0099 #bDoWrite := TRUE; 0100 END_IF; 0101 0102 #bReqActive := TRUE; 0103 END_IF; 0104 0105 // Start the request 0106 #PNUWrite(REQ := #bDoWrite, 0107 ID := #iID, 0108 INDEX := 47, 0109 LEN := #wLenght, 0110 BUSY => #bPNUWriteBusy, 0111 DONE => #bPNUWriteDone, 0112 ERROR => #bPNUWriteError, 0113 STATUS => #dwPNUWriteStatus, 0114 RECORD := #byWriteArray); 0115 0116 // Read the answer 0117 #PNURead(REQ := #bDoRead, 0118 ID := #iID, 0119 INDEX := 47, 0120 MLEN := 148, 0121 BUSY => #bPNUReadBusy, 0122 VALID => #bPNUReadDone, 0123 ERROR => #bPNUReadError, 0124 STATUS => #dwPNUReadStatus, 0125 LEN => #wDataLenght, 0126 RECORD := #byReadArray); 0127 0128 // Copie the read data to the output 0129 IF NOT #iRequestType THEN 0130 // Set the data automatically according to the Format 0131 IF #byReadArray[4] = 16#41 THEN 0132 #ioData.%B0 := #byReadArray[6]; // Byte 10 Value 0133 #ioData.%B1 := 0; 0134 #ioData.%B2 := 0; 0135 #ioData.%B3 := 0; 0136 ELSE 0137 IF #byReadArray[4] = 16#42 THEN</pre>		

Totally Integrated Automation Portal

```
0138      #ioData.%B0 := #byReadArray[7];    // Byte 12&13 Value
0139      #ioData.%B1 := #byReadArray[6];    // Byte 12&13 Value
0140      #ioData.%B2 := 0;
0141      #ioData.%B3 := 0;
0142      ELSE
0143      IF #byReadArray[4] = 16#43 THEN
0144          #ioData.%B0 := #byReadArray[9];    // Byte 12&13 Value
0145          #ioData.%B1 := #byReadArray[8];    // Byte 12&13 Value
0146          #ioData.%B2 := #byReadArray[7];    // Byte 14&15 Value
0147          #ioData.%B3 := #byReadArray[6];    // Byte 14&15 Value
0148      END_IF;
0149      END_IF;
0150      END_IF;
0151  END_IF;
0152
0153  // Reset the request bit for the read and write
0154  IF #bPNUWriteBusy THEN
0155      #bDoWrite := FALSE;
0156      #bWriteActive := TRUE;
0157  ELSIF #bPNUWriteDone THEN
0158      #bDoRead := TRUE;
0159      #bWriteActive := FALSE;
0160  END_IF;
0161  IF #bPNUReadBusy THEN
0162      #bDoRead := FALSE;
0163      #bReadActive := TRUE;
0164  ELSIF #bPNUReadDone THEN
0165      #bReadActive := FALSE;
0166  END_IF;
0167
0168  // Set the output bit after a request is processed
0169  IF #oActive AND #byReadArray[0] = #byReference THEN
0170      #oActive := FALSE;
0171      #oDone := TRUE;
0172  END_IF;
0173
0174  // Check for the rising edge of req
0175  IF NOT #iStartRequest AND #bReqActive THEN
0176      #bReqActive := FALSE;
0177  END_IF;
0178
```

Symbol	Address	Type	Comment
#bDoRead		Bool	
#bDoWrite		Bool	
#bPNUReadBusy		Bool	
#bPNUReadDone		Bool	
#bPNUReadError		Bool	
#bPNUWriteBusy		Bool	
#bPNUWriteDone		Bool	
#bPNUWriteError		Bool	
#bReadActive		Bool	
#bReqActive		Bool	
#bWriteActive		Bool	
#byReadArray		Array	
#byReference		Byte	
#byWriteArray		Array	
#dwPNUReadStatus		DWord	
#dwPNUWriteStatus		DWord	
#iFormat		Int	Datatype 0x41 = Byte, 0x42 = Word, 0x43 = DWord
#iID		HW_IO	unique connection ID
#ioData		DWord	
#ioData.%B0		Byte	
#ioData.%B1		Byte	
#ioData.%B2		Byte	
#ioData.%B3		Byte	
#iPNUnumber.%B0		Byte	PNU address
#iPNUnumber.%B1		Byte	PNU address
#iPNUsubIndex.%B0		Byte	PNU Subindex
#iPNUsubIndex.%B1		Byte	PNU Subindex
#iRequestType		Bool	0 = Read; 1 = Write
#iStartRequest		Bool	start Read or Write
#oActive		Bool	request in progress
#oDone		Bool	request finished
#oError		Bool	request failed
#oStatus		DWord	status of the failed function
#PNURead		Multi_SFB	
#PNUWrite		Multi_SFB	
#wDataLenght		Word	
#wLenght		Word	