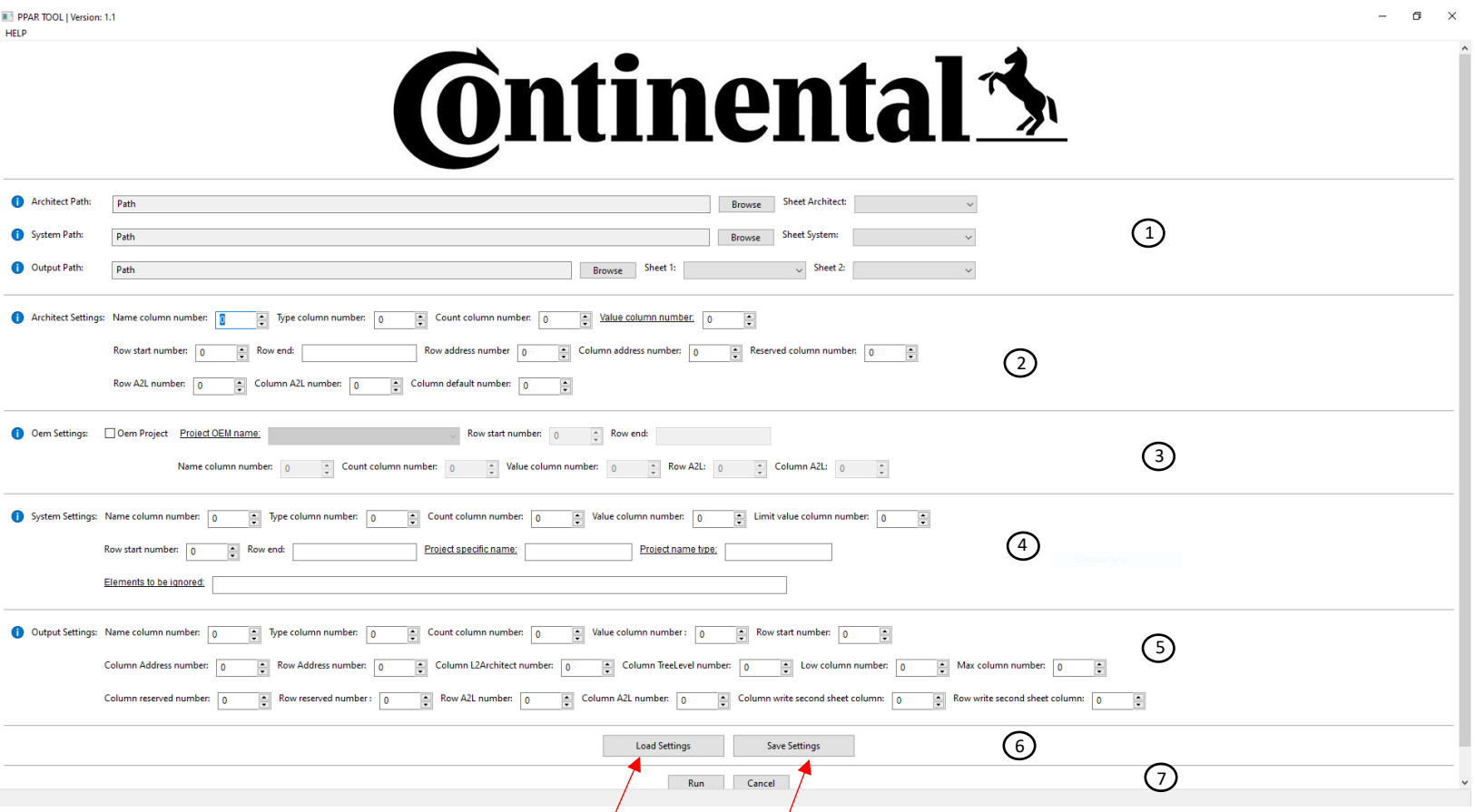
Why the PPAR Generator Tool exists?

To generate PPAR .mot and .a2l files for one project a manual fill in of data needs to be made from 2 Excels files provided by architect and system responsible. This manual process is obviously prone to human errors that will impact production parameters generation.

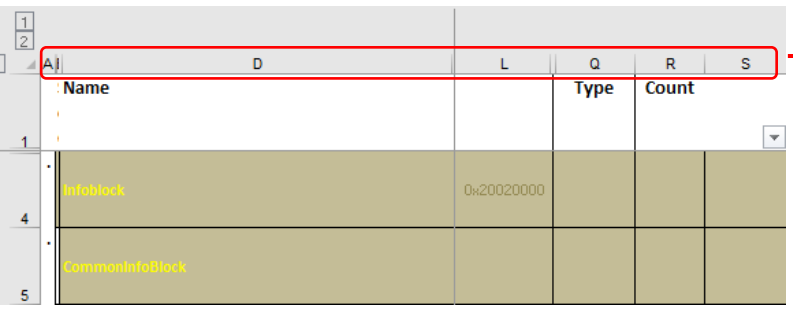
The purpose of this tool is that automatically analysis the types, size, values of each parameters from both Excels. The not matching information is signaled. The tool is correctly merging the 2 system and architecture files into another Excel that is used to generate the .mot and .a2l files.

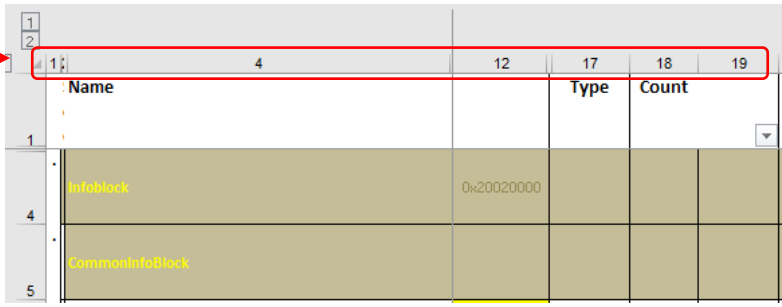


As we can see, the interface is a large interface with loads of inputs needed for that Tool to work properly, fortunate, the **Load Settings** and the **Save Settings** buttons are there to solve this problem. Most inputs will be the same from one project to another, but there needs to be all the inputs on the GUI to accomplish importance of the Future Proof.

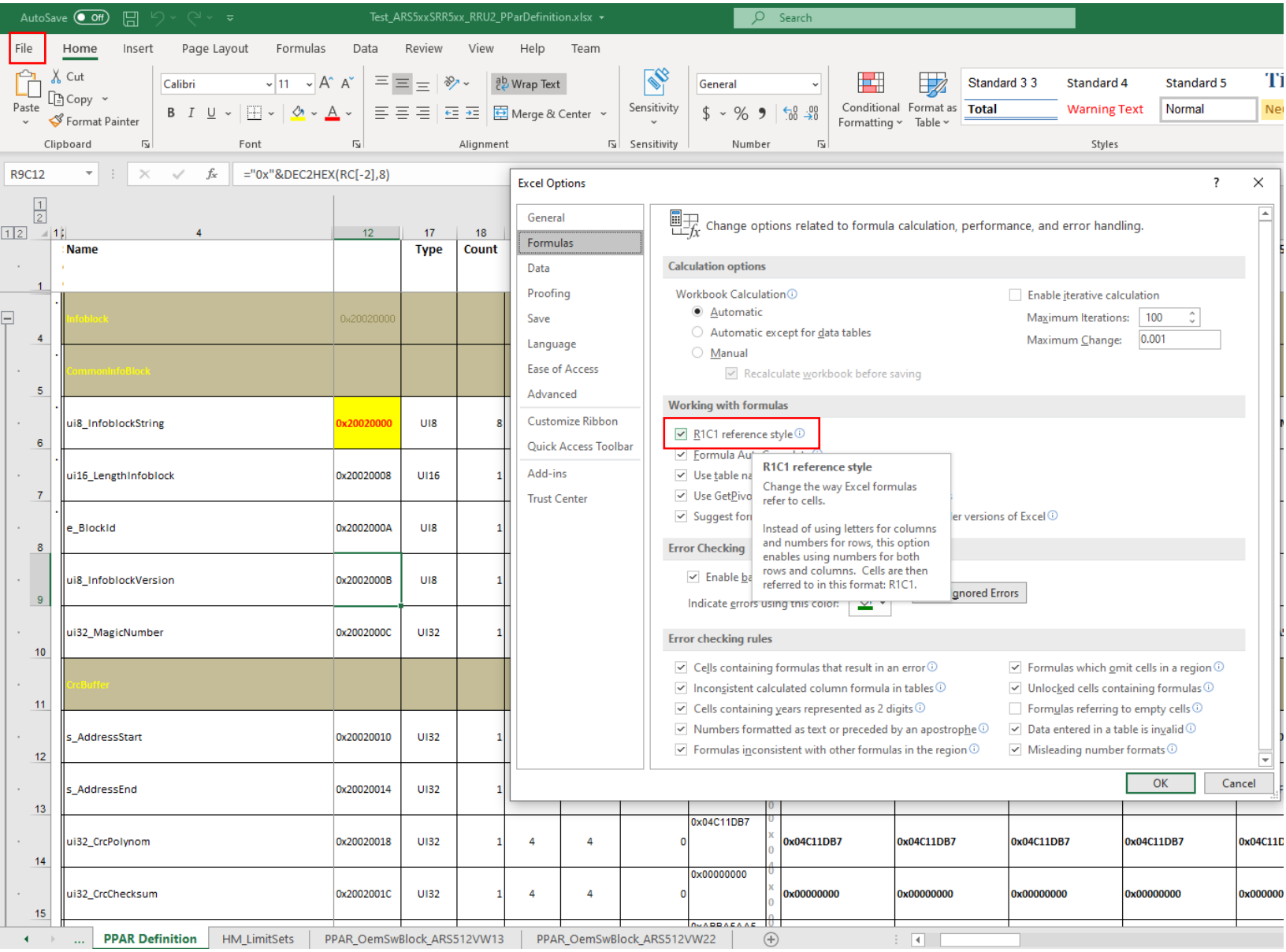
Pressing the Load Settings button will send the user to select a .txt file with all the data needed for section (2) to (5). Pressing the Save Settings button will send the user to enter the name of the .txt file with all the data from the currently interface from section (2) to (5).

You will need to set **R1C1 reference style** in the excels to help you easily see the number of columns instead of letters.

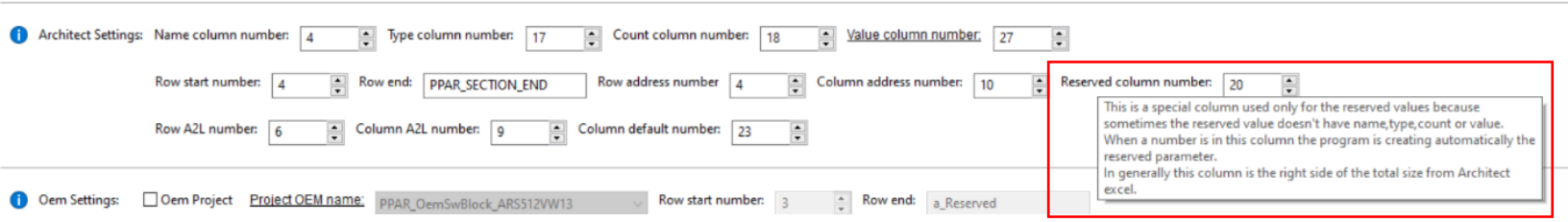


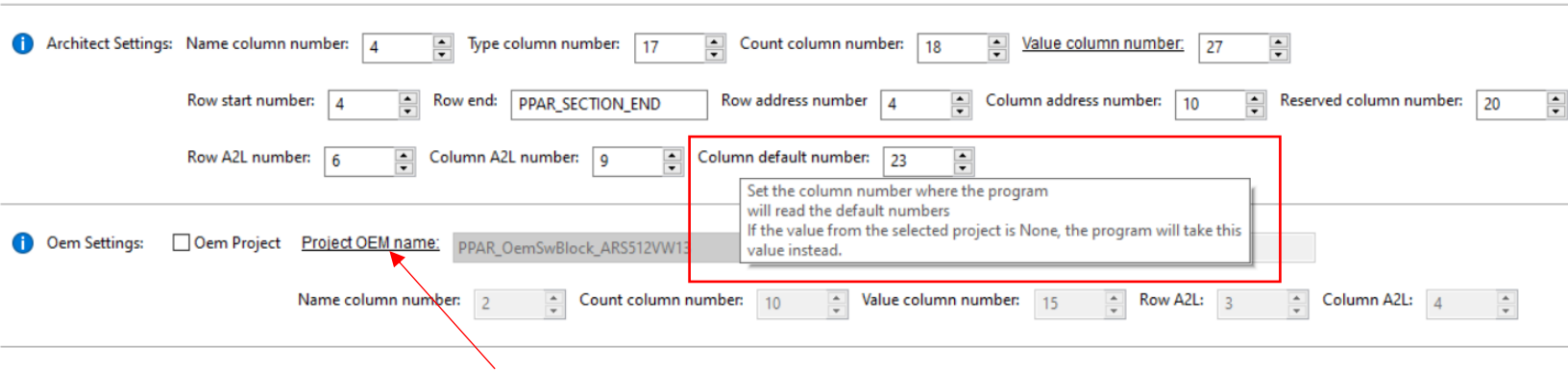


File -> Options -> Formulas -> R1C1 reference style



Every single static text from the GUI has more information about the selected parameter. If the user hovers the text, information will appear and its very helpful, the user doesn't need to search the documentation for only one parameter.





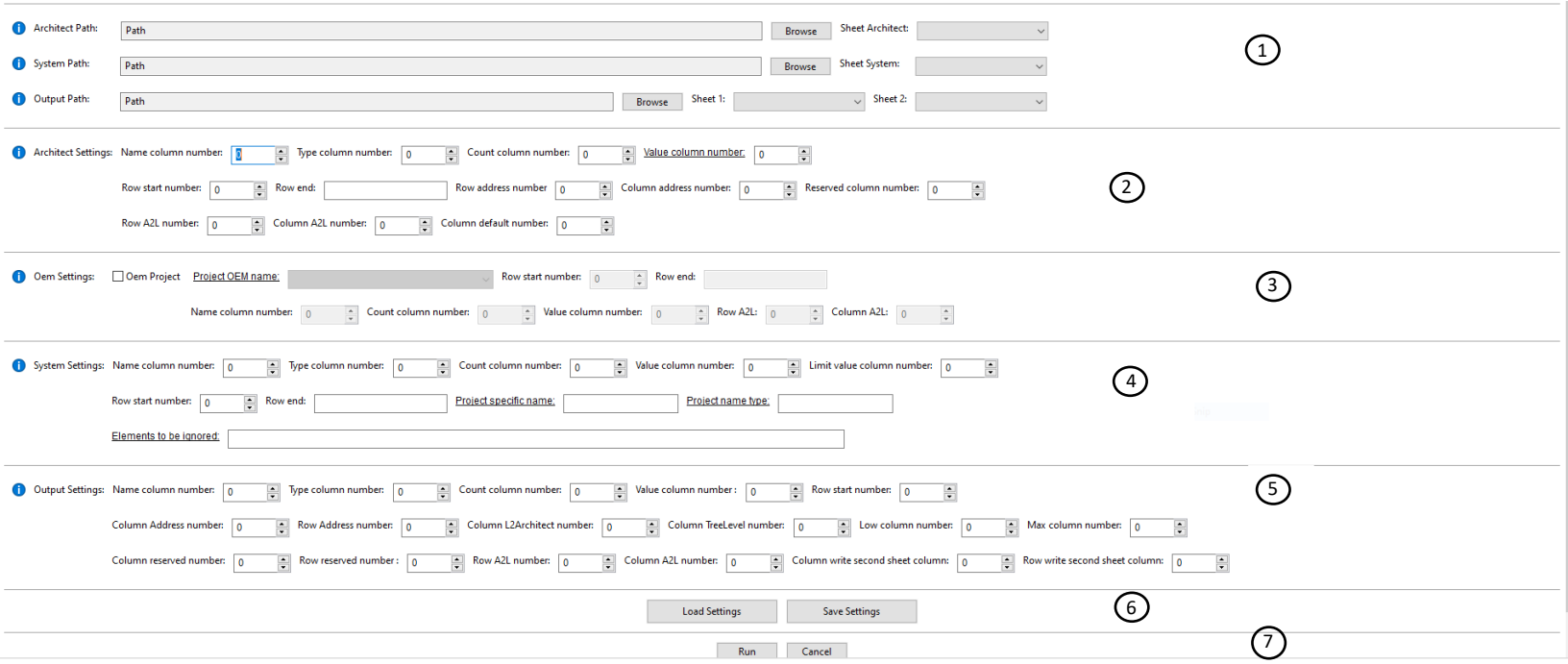
Some text is <u>underlined</u> in the interface, these values need to be updated from one project to another. These values are very important to be verified before the start of the program and double checked!

How to use the PPAR Generator Tool?

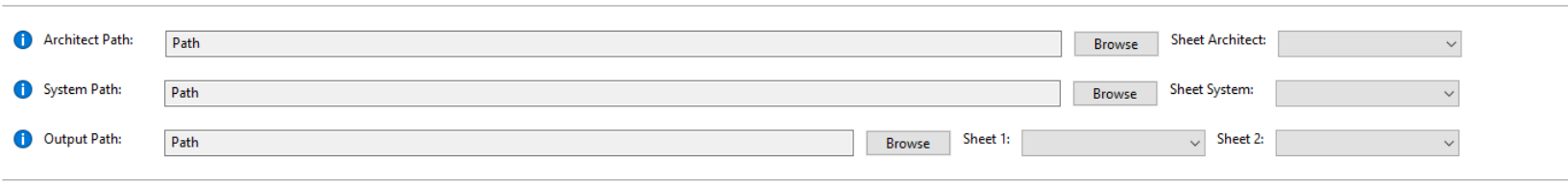
The user interface is fragmented in separately sections:

1. Path section.
2. Architect Settings.
3. OEM Settings.
4. System Settings.
5. Output Settings.
6. Load and Save Settings Buttons.
7. Run and Cancel Buttons.

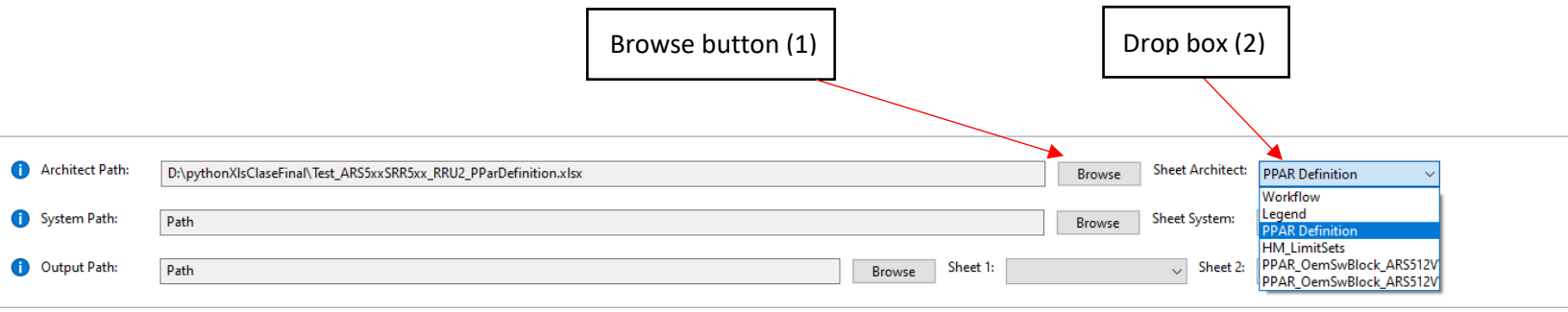
Let's take every section individually and explain what these means for the user.



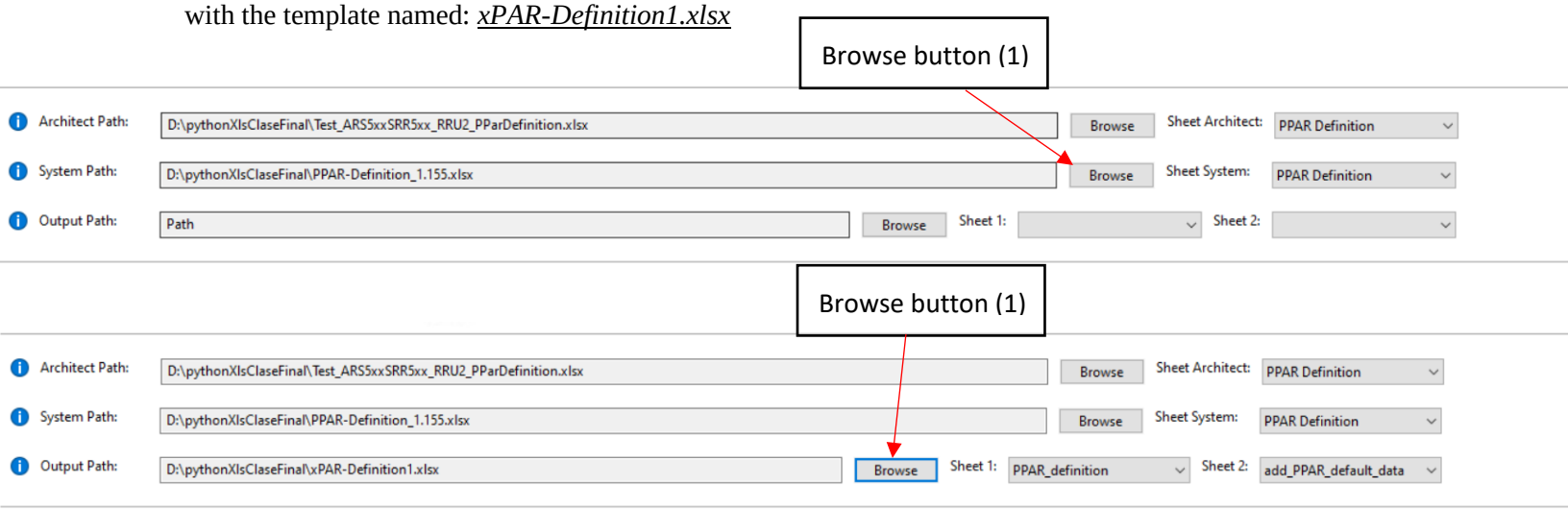
1. Path section.



Firstly, we need to select the Architect excel by pressing the *Browse* button (1). This excel can be anywhere in your personal computer, the Tool will work with the path given. The sheet that the program needs to read data is selected automatically, but the user can choose different sheet if needed from the *drop box (2).* The content of the drop box loads after the Architect excel is loaded.

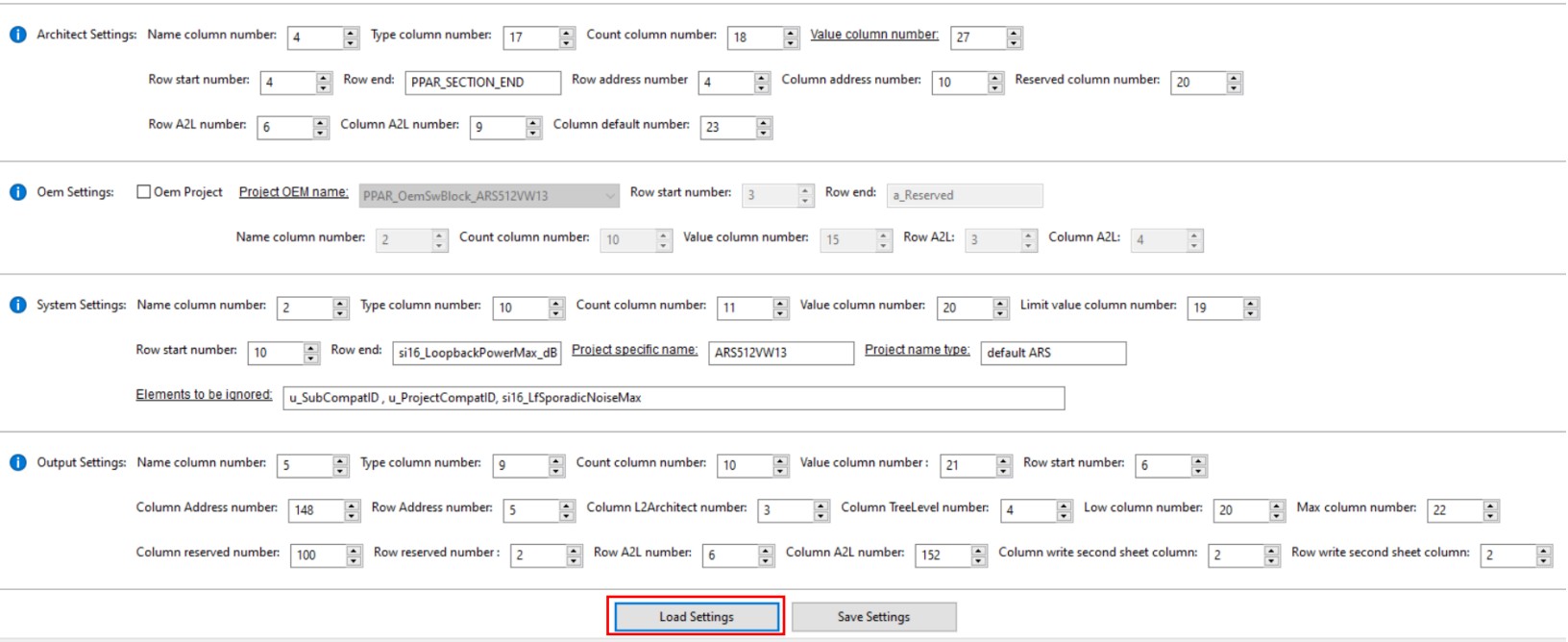


Repeat the first step with the System excel and for the Output excel. For the Output select the Excel with the template named: *xPAR-Definition1.xlsx*

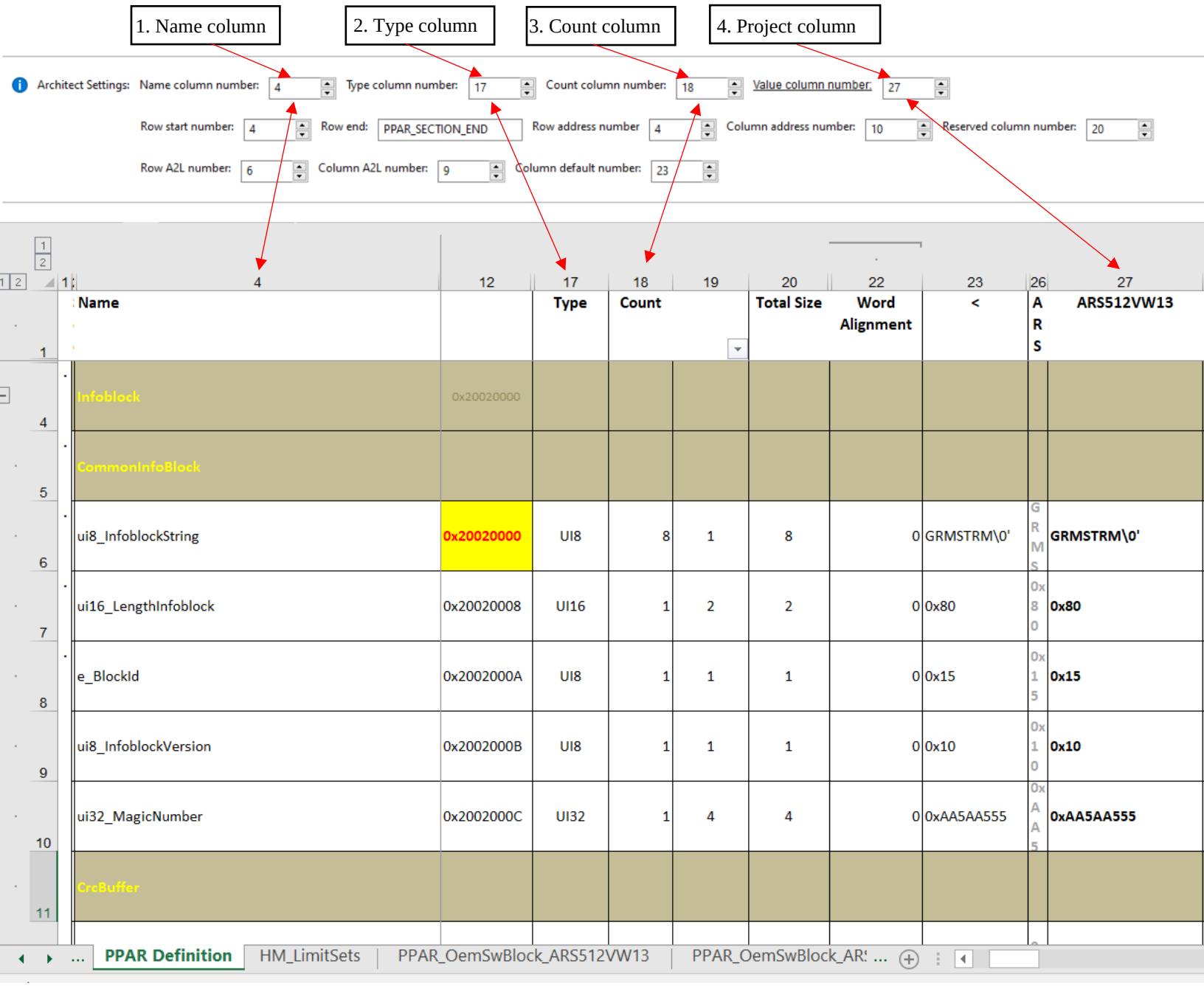


6. Load Settings and Save Settings Section

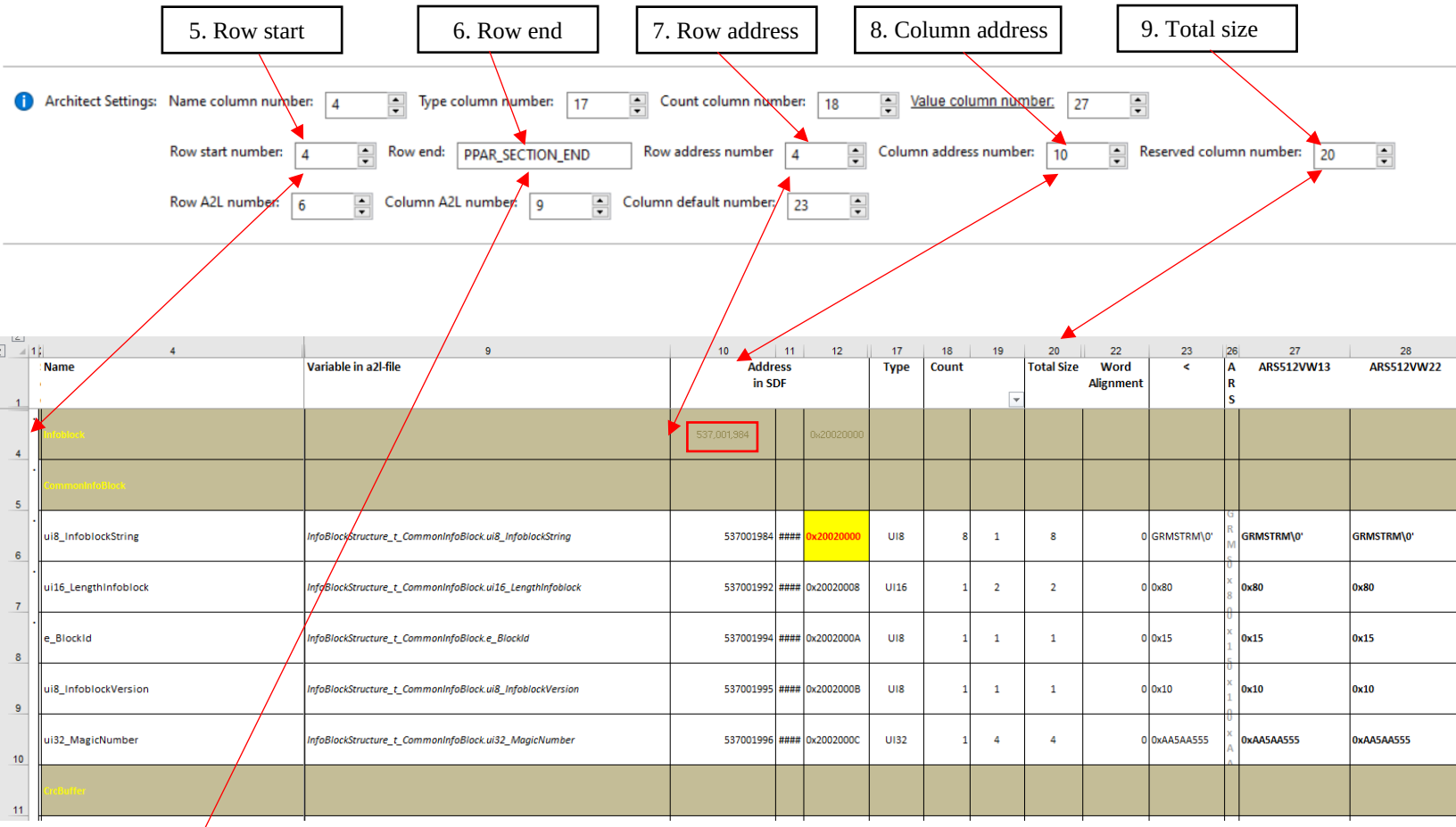
To save time and to understand the sections (2) - (5) let's load one .txt file *RRUR_ars512VW13.txt* by pressing the Load Settings Button. This .txt file has all the interface inputs for the project *ARS512VW1*.



2. Architect Settings Section



1. Set the column number from where the program will read names of the parameters.
2. Set the column number from where the program will read type of the parameters.
3. Set the column number from where the program will read count of the parameters.
4. ***Set the column number from where the program will read values of the parameters of the wanted project. Depending on the wanted project, this column number will need to be set manually. The column number 27 is corresponding to the project ARS512VW13!**



	1	4	9	10	11	12	17	18	19	20	22	23	26	27	28
		Name	Variable in a2l-file	Address in SDF			Type	Count		Total Size	Word Alignment	<	ARS	ARS512VW13	ARS512VW22
4		Infoblock		537,001,984		0x20020000									
5		CommonInfoBlock													
6		ui8_InfoblockString	*InfoBlockStructure_t_CommonInfoBlock.ui8_InfoblockString*	537001984	####	0x20020000	UI8	8	1	8	0	GRMSTRM\0'	GRM	GRMSTRM\0'	GRMSTRM\0'
7		ui16_LengthInfoblock	*InfoBlockStructure_t_CommonInfoBlock.ui16_LengthInfoblock*	537001992	####	0x20020008	UI16	1	2	2	0	0x80	0x8	0x80	0x80
8		e_BlockId	*InfoBlockStructure_t_CommonInfoBlock.e_BlockId*	537001994	####	0x2002000A	UI8	1	1	1	0	0x15	0x15	0x15	0x15
9		ui8_InfoblockVersion	*InfoBlockStructure_t_CommonInfoBlock.ui8_InfoblockVersion*	537001995	####	0x2002000B	UI8	1	1	1	0	0x10	0x10	0x10	0x10
10		ui32_MagicNumber	*InfoBlockStructure_t_CommonInfoBlock.ui32_MagicNumber*	537001996	####	0x2002000C	UI32	1	4	4	0	0xAA5AA555	0xAA	0xAA5AA555	0xAA5AA555
11		CrcBuffer													

...

727	PPAR Serial Nb	0x20023FB0	UI8	26	1	26
728	Trace Nb	0x20023FCA	UI8	38	1	38
729	PPAR_SECTION_END			0	0	

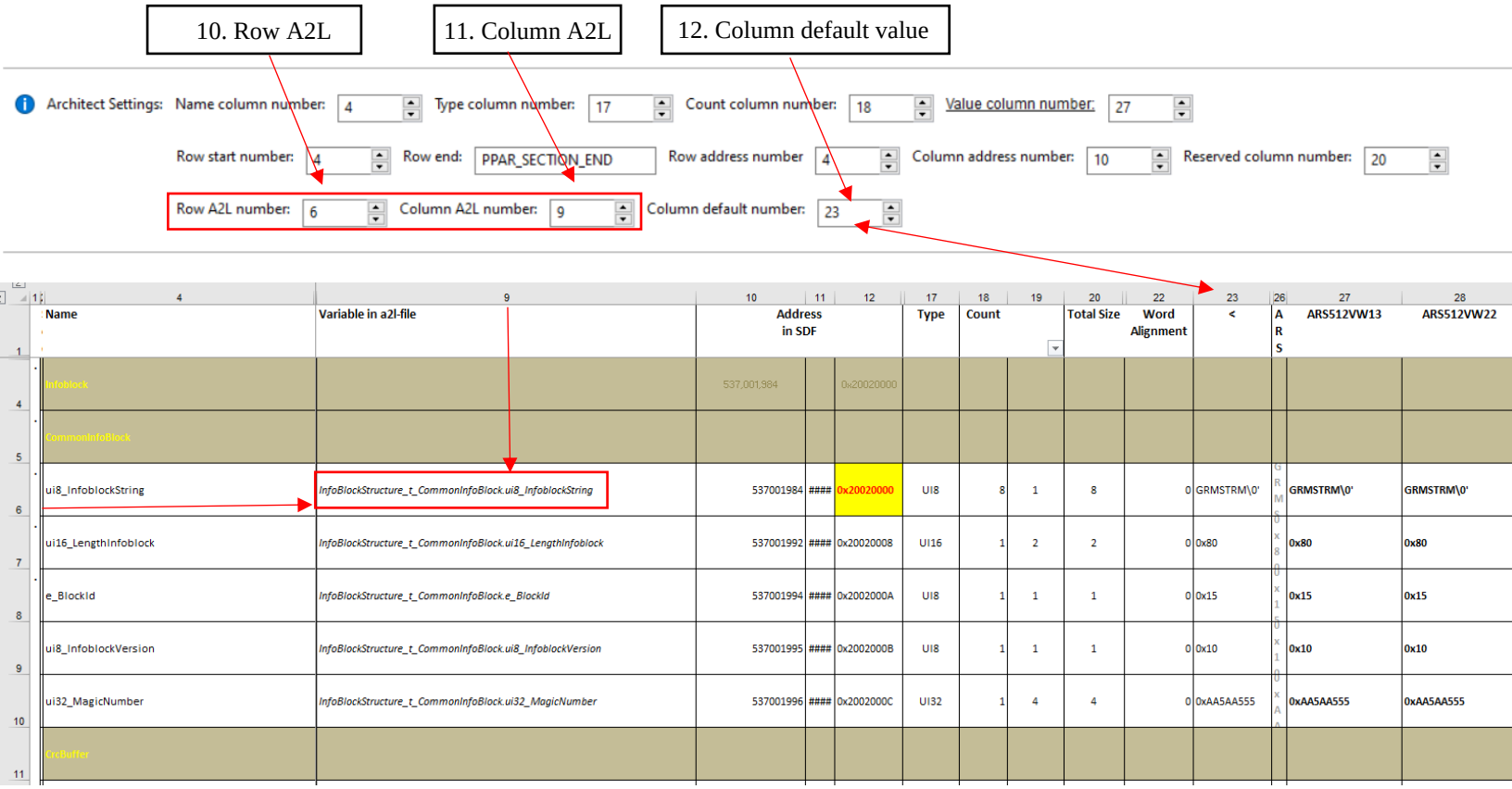
5. Set the row number from where the program will start reading data.

6. Set the last name of the excel to stop reading data. This parameter won't be in the output, this is just a red line to stop reading.

7. Set the row number where the starting address is.

8. Set the column number where the starting address is, in decimal.

9. Set the column number from where the program will read how much of the memory occupy by the parameter. This number is calculated by the formula *Count * bytes*.



	Name	Variable in a2l-file	Address in SDF			Type	Count		Total Size	Word Alignment	<	ARS	ARS512VW13	ARS512VW22
4	InfoBlock		537,001,984		0x20020000									
5	CommonInfoBlock													
6	ui8_InfoblockString	*InfoBlockStructure_t_CommonInfoBlock.ui8_InfoblockString*	537001984	####	0x20020000	UI8	8	1	8	0	GRMSTRM\0'	GRM	GRMSTRM\0'	GRMSTRM\0'
7	ui16_LengthInfoblock	*InfoBlockStructure_t_CommonInfoBlock.ui16_LengthInfoblock*	537001992	####	0x20020008	UI16	1	2	2	0	0x80	0x8	0x80	0x80
8	e_BlockId	*InfoBlockStructure_t_CommonInfoBlock.e_BlockId*	537001994	####	0x2002000A	UI8	1	1	1	0	0x15	0x1	0x15	0x15
9	ui8_InfoblockVersion	*InfoBlockStructure_t_CommonInfoBlock.ui8_InfoblockVersion*	537001995	####	0x2002000B	UI8	1	1	1	0	0x10	0x1	0x10	0x10
10	ui32_MagicNumber	*InfoBlockStructure_t_CommonInfoBlock.ui32_MagicNumber*	537001996	####	0x2002000C	UI32	1	4	4	0	0xAA5AA555	0xAA	0xAA5AA555	0xAA5AA555
11	SrcBuffer													

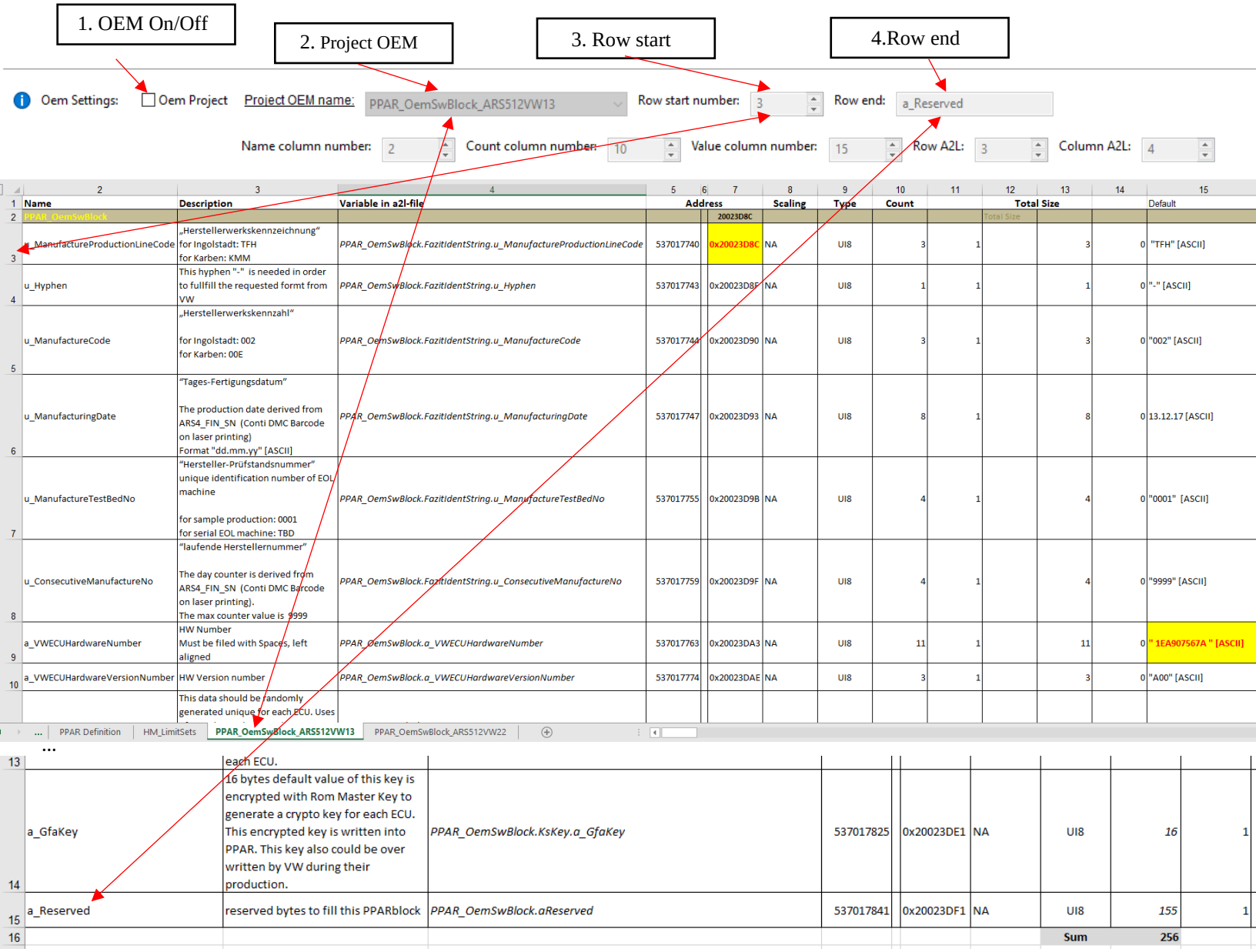
10. Set the row number where the first a2l value is.

11. Set the column number where a2l values are.

12. Set the column number where are the default values. If a parameter doesn't have written a value in the selected project, the value will be set from this column. Example:

	Name	Variable in a2l-file	Address in SDF			Type	Count		Total Size	Word Alignment	<	ARS	ARS512VW13
105	HM_Country_Selector	PPAR_Homologation.HM_CountrySelector	537004016	####	0x200207F0	UI8	1	1	1	1	0xFFFF		
106	align1	PPAR_Homologation.align1	537004017	####	0x200207F1	UI8	3	1	3	0	0xFFFF		0xFFFFFF00
107	HM_allowCOM	PPAR_Homologation.HM_AllowCOM	537004020	####	0x200207F4	UI32	1	4	4	0	0xFFFF		

3. OEM Settings Section

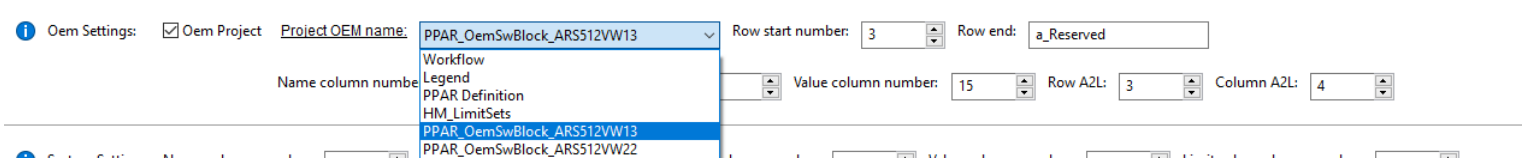


Name	Description	Variable in a2l-file	Address		Scaling	Type	Count		Total Size			Default
PPAR_OemSwBlock				20023D8C								
u_ManufactureProductionLineCode	„Herstellerwerkskennzeichnung“ for Ingolstadt: TFH for Karben: KMM	*PPAR_OemSwBlock.FazitIdentString.u_ManufactureProductionLineCode*	537017740	0x20023D8C	NA	UI8	3	1		3	0	"TFH" [ASCII]
u_Hyphen	This hyphen "-" is needed in order to fullfill the requested formt from VW	*PPAR_OemSwBlock.FazitIdentString.u_Hyphen*	537017743	0x20023D8F	NA	UI8	1	1		1	0	"-" [ASCII]
u_ManufactureCode	„Herstellerwerkskennzahl“ for Ingolstadt: 002 for Karben: 00E	*PPAR_OemSwBlock.FazitIdentString.u_ManufactureCode*	537017744	0x20023D90	NA	UI8	3	1		3	0	"002" [ASCII]
u_ManufacturingDate	“Tages-Fertigungsdatum” The production date derived from ARS4_FIN_SN (Conti DMC Barcode on laser printing) Format "dd.mm.yy" [ASCII]	*PPAR_OemSwBlock.FazitIdentString.u_ManufacturingDate*	537017747	0x20023D93	NA	UI8	8	1		8	0	13.12.17 [ASCII]
u_ManufactureTestBedNo	“Hersteller-Prüfstandsnummer” unique identification number of EOL machine for sample production: 0001 for serial EOL machine: TBD	*PPAR_OemSwBlock.FazitIdentString.u_ManufactureTestBedNo*	537017755	0x20023D9B	NA	UI8	4	1		4	0	"0001" [ASCII]
u_ConsecutiveManufactureNo	“laufende Herstellernummer” The day counter is derived from ARS4_FIN_SN (Conti DMC Barcode on laser printing). The max counter value is 9999	*PPAR_OemSwBlock.FazitIdentString.u_ConsecutiveManufactureNo*	537017759	0x20023D9F	NA	UI8	4	1		4	0	"9999" [ASCII]
a_VWECUHardwareNumber	HW Number Must be filed with Spaces, left aligned	*PPAR_OemSwBlock.a_VWECUHardwareNumber*	537017763	0x20023DA3	NA	UI8	11	1		11	0	" 1EA907567A " [ASCII]
a_VWECUHardwareVersionNumber	HW Version number	*PPAR_OemSwBlock.a_VWECUHardwareVersionNumber*	537017774	0x20023DAE	NA	UI8	3	1		3	0	"A00" [ASCII]
	This data should be randomly generated unique for each ECU. Uses											

PPAR Definition | HM_LimitSets | PPAR_OemSwBlock_ARS512VW13 | PPAR_OemSwBlock_ARS512VW22

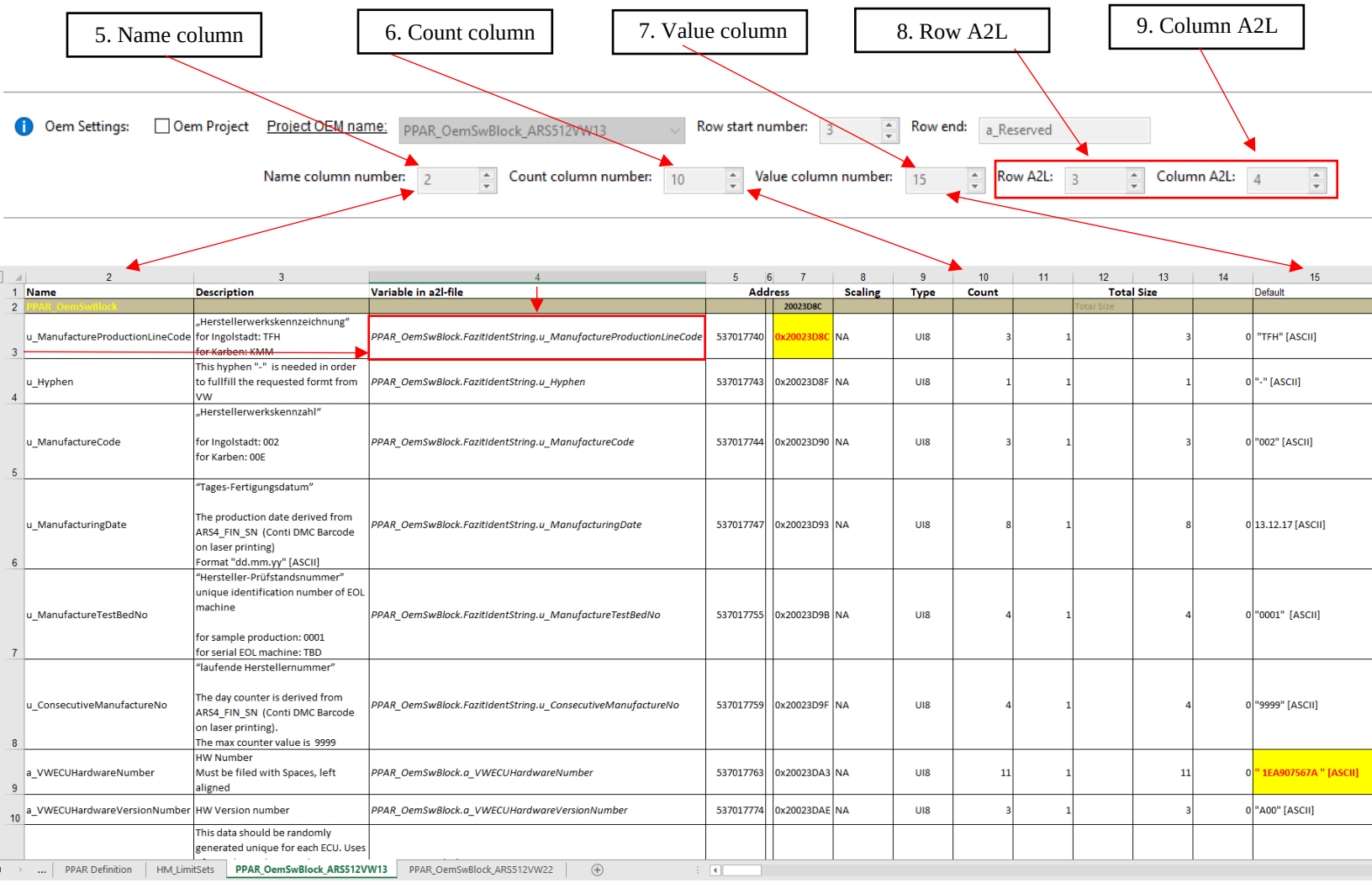
	each ECU.									
a_GfaKey	16 bytes default value of this key is encrypted with Rom Master Key to generate a crypto key for each ECU. This encrypted key is written into PPAR. This key also could be over written by VW during their production.	*PPAR_OemSwBlock.KsKey.a_GfaKey*	537017825	0x20023DE1	NA	UI8	*16*	1		
a_Reserved	reserved bytes to fill this PPARblock	*PPAR_OemSwBlock.aReserved*	537017841	0x20023DF1	NA	UI8	*155*	1		
							Sum	256		

1. Check the box if the project has OEM sheet.
2. ***Select the OEM project from drop box. This filed is important and will be different from one project to another.**



3. Set the row number from where the program will start reading data

4. Set the last name of the excel to stop reading data. This parameter won't be in the output, this is just a red line to stop reading.



	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Name	Description	Variable in a2l-file	Address			Scaling	Type	Count		Total Size			Default
2	PPAR_OemSwBlock					20023D8C					Total Size			
3	u_ManufactureProductionLineCode	„Herstellerwerkskennzeichnung“ for Ingolstadt: TFH for Karben: KMM	PPAR_OemSwBlock.FazitIdentString.u_ManufactureProductionLineCode	537017740		0x20023D8C	NA	UI8	3	1		3	0	"TFH" [ASCII]
4	u_Hyphen	This hyphen "-" is needed in order to fullfill the requested formt from VW	PPAR_OemSwBlock.FazitIdentString.u_Hyphen	537017743		0x20023D8F	NA	UI8	1	1		1	0	"-" [ASCII]
5	u_ManufactureCode	„Herstellerwerkskennzahl“ for Ingolstadt: 002 for Karben: 00E	PPAR_OemSwBlock.FazitIdentString.u_ManufactureCode	537017744		0x20023D90	NA	UI8	3	1		3	0	"002" [ASCII]
6	u_ManufacturingDate	“Tages-Fertigungsdatum” The production date derived from ARS4_FIN_SN (Conti DMC Barcode on laser printing) Format "dd.mm.yy" [ASCII]	PPAR_OemSwBlock.FazitIdentString.u_ManufacturingDate	537017747		0x20023D93	NA	UI8	8	1		8	0	13.12.17 [ASCII]
7	u_ManufactureTestBedNo	“Hersteller-Prüfstandsnummer” unique identification number of EOL machine for sample production: 0001 for serial EOL machine: TBD	PPAR_OemSwBlock.FazitIdentString.u_ManufactureTestBedNo	537017755		0x20023D9B	NA	UI8	4	1		4	0	"0001" [ASCII]
8	u_ConsecutiveManufactureNo	“laufende Herstellernummer” The day counter is derived from ARS4_FIN_SN (Conti DMC Barcode on laser printing). The max counter value is 9999	PPAR_OemSwBlock.FazitIdentString.u_ConsecutiveManufactureNo	537017759		0x20023D9F	NA	UI8	4	1		4	0	"9999" [ASCII]
9	a_VWECUHardwareNumber	HW Number Must be filed with Spaces, left aligned	PPAR_OemSwBlock.a_VWECUHardwareNumber	537017763		0x20023DA3	NA	UI8	11	1		11	0	" 1EA907567A " [ASCII]
10	a_VWECUHardwareVersionNumber	HW Version number	PPAR_OemSwBlock.a_VWECUHardwareVersionNumber	537017774		0x20023DAE	NA	UI8	3	1		3	0	"A00" [ASCII]
		This data should be randomly generated unique for each ECU. Uses												

PPAR Definition | HM_LimitSets | PPAR_OemSwBlock_ARS512VW13 | PPAR_OemSwBlock_ARS512VW22

5. Set the column number from where the program will read names of the parameters.

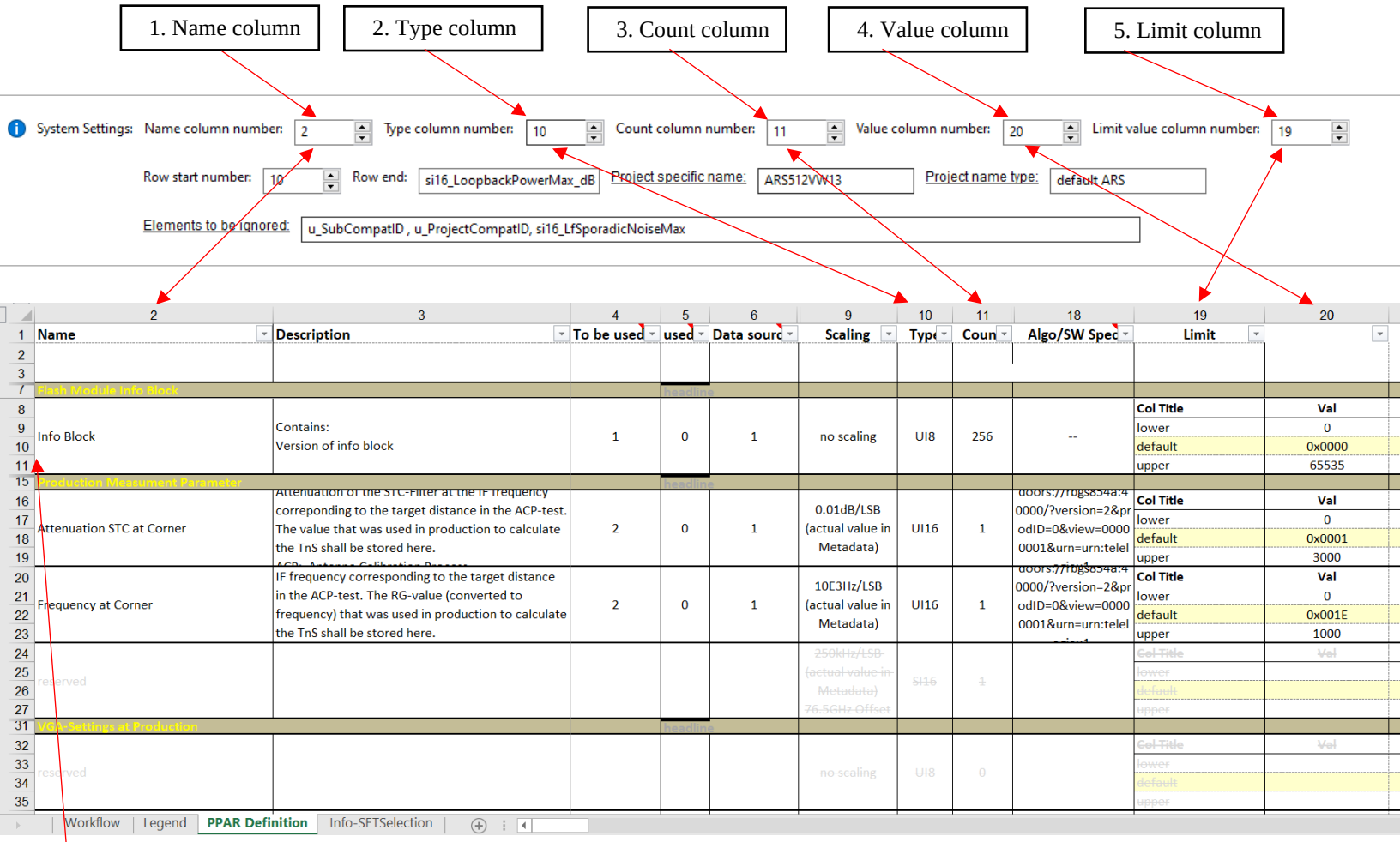
6. Set the column number from where the program will read count of the parameters.

7. Set the column number from where the program will read values of the parameters.

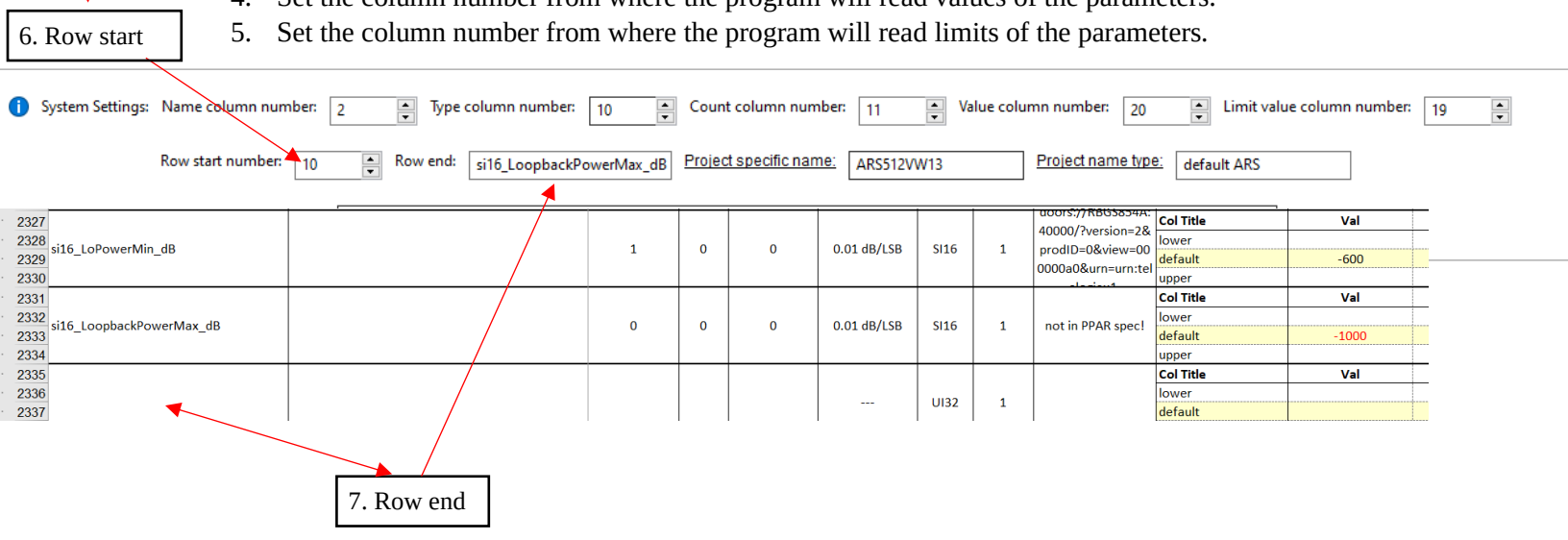
8. Set the row number where the first a2l value is.

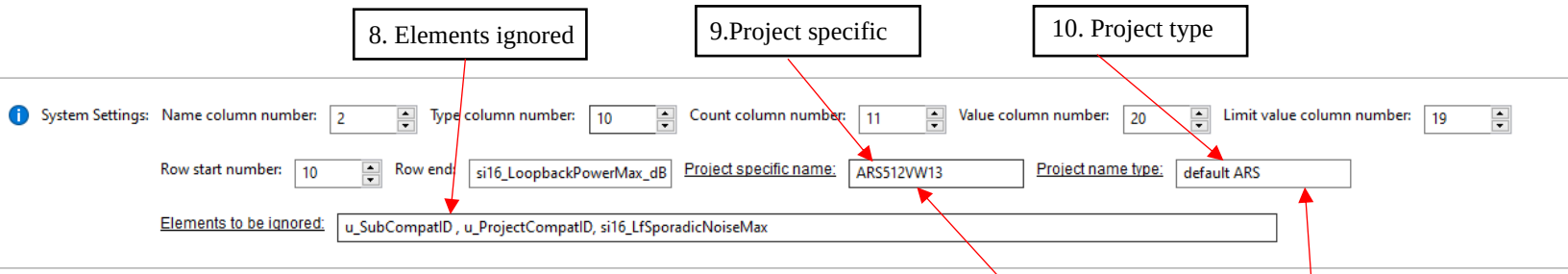
9. Set the column number where a2l values are.

4. System Settings Section



1. Set the column number from where the program will read names of the parameters.
2. Set the column number from where the program will read type of the parameters.
3. Set the column number from where the program will read count of the parameters.
4. Set the column number from where the program will read values of the parameters.
5. Set the column number from where the program will read limits of the parameters.





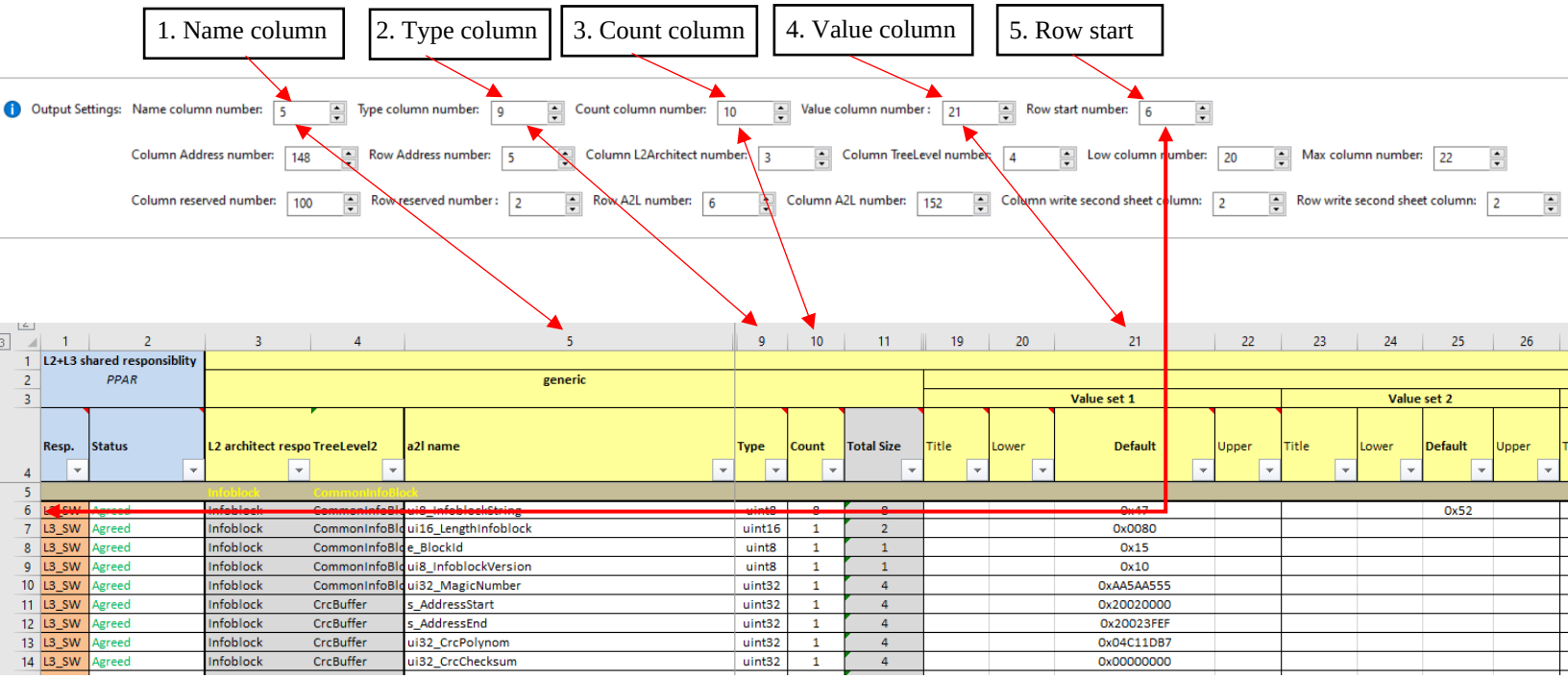
6. Set the row number from where the program will start reading data.
7. Set the last name of the excel to stop reading data. This parameter will be read, and its value will update the value from Architect.
8. ***Set the name of the parameter that doesn't need update. These values will be ignored from the System excel. In the output these parameters will have the values from Architect Excel.**
9. ***Set the project specific name, some projects have values specific by the project:**

										Col Title	Val
2227										lower	
2228										default	97
2229	f_MmicNeNo0_UL	NoEmission --> NormalOperation (VCO temp. sensor)	1	RfCom	0	°C	float	1	doors://RBGS854A:40000/?version=2&prodID=0&view=0000a0&urn=urn:telelogic::1-503e822e5ec3651e-O-5917-00041866	default SRR520HA10	120
										SRR520LD10	
										/ARS510HA10	
										ARS510LD10	
										ARS510RT10	
										ARS511VP12	
2230										SRR521BW12	
2231										upper	

10. ***Set the project name type, the projects have different values:**

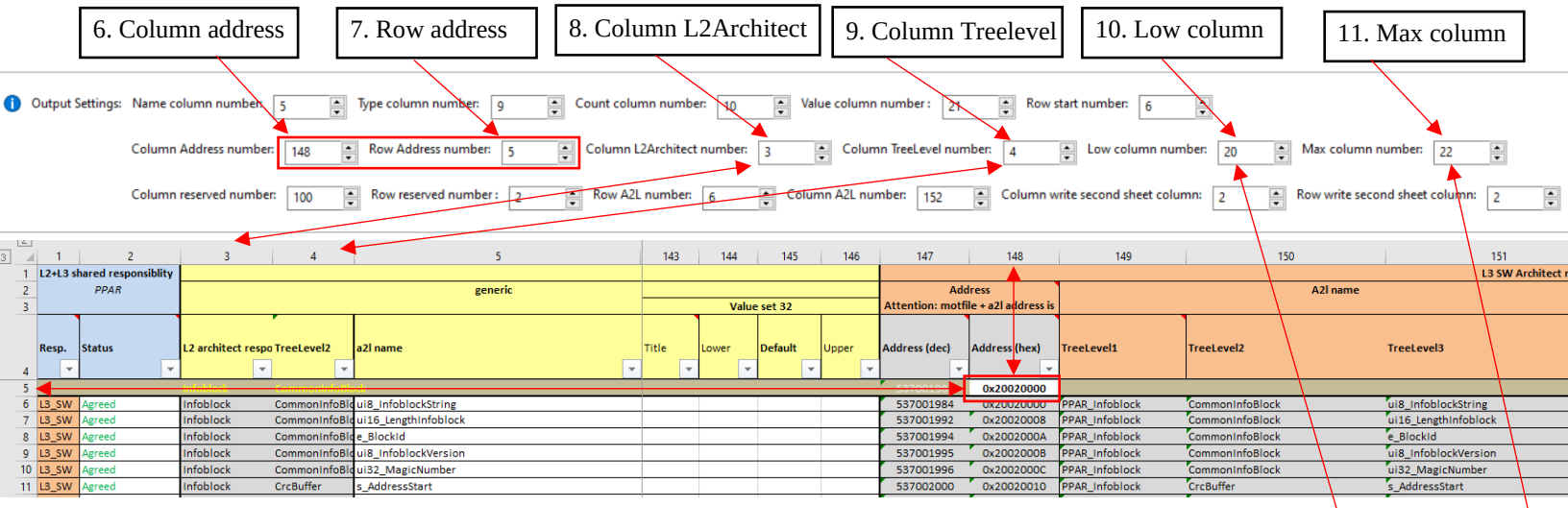
										Col Title	Val
1641									doors://RBGS854A:		
1642									40000/?version=2&	lower	
1643	HM_LimitSets_CRC	CRC used by Homologation Monitor (HM) to check if LimitSets section of PPAR did not change	1	HoMon	0	---	UI32	1	prodID=0&view=00	default ARS	0xA371DE65
1644									000003&urn=urn:tel	default SRR	0xA704A492
1645									elogic::1-	upper	

5. Output Section



	1	2	3	4	5	9	10	11	19	20	21	22	23	24	25	26
1	L2+L3 shared responsiblity															
2	PPAR		generic													
3									Value set 1				Value set 2			
4	Resp.	Status	L2 architect respo	TreeLevel2	a2l name	Type	Count	Total Size	Title	Lower	Default	Upper	Title	Lower	Default	Upper
5			Infoblock	CommonInfoBlock												
6	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui8_InfoblockString	uint8	8	8			0x47				0x52	
7	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui16_LengthInfoblock	uint16	1	2			0x0080					
8	L3_SW	Agreed	Infoblock	CommonInfoBlo	e_BlockId	uint8	1	1			0x15					
9	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui8_InfoblockVersion	uint8	1	1			0x10					
10	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui32_MagicNumber	uint32	1	4			0xAA5AA555					
11	L3_SW	Agreed	Infoblock	CrcBuffer	s_AddressStart	uint32	1	4			0x20020000					
12	L3_SW	Agreed	Infoblock	CrcBuffer	s_AddressEnd	uint32	1	4			0x20023FEF					
13	L3_SW	Agreed	Infoblock	CrcBuffer	ui32_CrcPolynom	uint32	1	4			0x04C11DB7					
14	L3_SW	Agreed	Infoblock	CrcBuffer	ui32_CrcChecksum	uint32	1	4			0x00000000					

1. Set the column number where the program will write names of the parameters.
2. Set the column number where the program will write type of the parameters.
3. Set the column number where the program will write count of the parameters.
4. Set the column number where the program will write count of the parameters.
5. Set the row number from where the program will start writing data.



	1	2	3	4	5	143	144	145	146	147	148	149	150	151
1	L2+L3 shared responsiblity													L3 SW Architect r
2	PPAR		generic				Value set 32			Address		A2l name		
3										Attention: motfile + a2l address is				
4	Resp.	Status	L2 architect respo	TreeLevel2	a2l name	Title	Lower	Default	Upper	Address (dec)	Address (hex)	TreeLevel1	TreeLevel2	TreeLevel3
5			Infoblock	CommonInfoBlock						537001984	0x20020000			
6	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui8_InfoblockString					537001984	0x20020000	PPAR_Infoblock	CommonInfoBlock	ui8_InfoblockString
7	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui16_LengthInfoblock					537001992	0x20020008	PPAR_Infoblock	CommonInfoBlock	ui16_LengthInfoblock
8	L3_SW	Agreed	Infoblock	CommonInfoBlo	e_BlockId					537001994	0x2002000A	PPAR_Infoblock	CommonInfoBlock	e_BlockId
9	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui8_InfoblockVersion					537001995	0x2002000B	PPAR_Infoblock	CommonInfoBlock	ui8_InfoblockVersion
10	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui32_MagicNumber					537001996	0x2002000C	PPAR_Infoblock	CommonInfoBlock	ui32_MagicNumber
11	L3_SW	Agreed	Infoblock	CrcBuffer	s_AddressStart					537002000	0x20020010	PPAR_Infoblock	CrcBuffer	s_AddressStart

	1	2	3	4	5	9	10	11	19	20	21	22
1	L2+L3 shared responsiblity											
2	PPAR		generic									
3									Value set 1			
4	Resp.	Status	L2 architect respo	TreeLevel2	a2l name	Type	Count	Total Size	Title	Lower	Default	Upper
122	L3_SW	Agreed	Calibartion	Calibration Val	CalSet1 RX1...4 setting	uint8	4	4		53	0xFF	53
123	L3_SW	Agreed	Calibartion	Calibration Val	align2	uint8	4	4			0x00	
124	L3_SW	Agreed	Calibartion	Calibration Val	CalSet2_AntCompValue	sint16	24	48		-32768	0x7FFF	32767
125	L3_SW	Agreed	Calibartion	Calibration Val	CalSet2 RX-Channel_Calibration	uint16	4	8		10361	0x7FFF	32767
126	L3_SW	Agreed	Calibartion	Calibration Val	CalSet2 TX1...3	uint8	3	3		0	0xFF	63

6. Set the column number where the program will write the first address.
7. Set the row number where the program will write the first address.
8. Set the column number where the program will write the section name. One section contains more subsections, this helps when the user is searching for parameters from one section/subsection.
9. Set the column number where the program will write the subblocks name.

	1	2	3	4	5	143	144	145	146	147	148	149	150	151
1	L2+L3 shared responsiblity													L3 SW Architect r
2	PPAR		generic							Address		A2l name		
3						Value set 32				Attention: motfile + a2l address is				
4	Resp.	Status	L2 architect respo	TreeLevel2	a2l name	Title	Lower	Default	Upper	Address (dec)	Address (hex)	TreeLevel1	TreeLevel2	TreeLevel3
5			Infoblock	CommonInfoBlock						537001984	0x20020000			
6	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui8_InfoblockString					537001984	0x20020000	PPAR_Infoblock	CommonInfoBlock	ui8_InfoblockString
7	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui16_LengthInfoblock					537001992	0x20020008	PPAR_Infoblock	CommonInfoBlock	ui16_LengthInfoblock
8	L3_SW	Agreed	Infoblock	CommonInfoBlo	e_BlockId					537001994	0x2002000A	PPAR_Infoblock	CommonInfoBlock	e_BlockId
9	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui8_InfoblockVersion					537001995	0x2002000B	PPAR_Infoblock	CommonInfoBlock	ui8_InfoblockVersion
10	L3_SW	Agreed	Infoblock	CommonInfoBlo	ui32_MagicNumber					537001996	0x2002000C	PPAR_Infoblock	CommonInfoBlock	ui32_MagicNumber
11	L3_SW	Agreed	Infoblock	CrcBuffer	s_AddressStart					537002000	0x20020010	PPAR_Infoblock	CrcBuffer	s_AddressStart

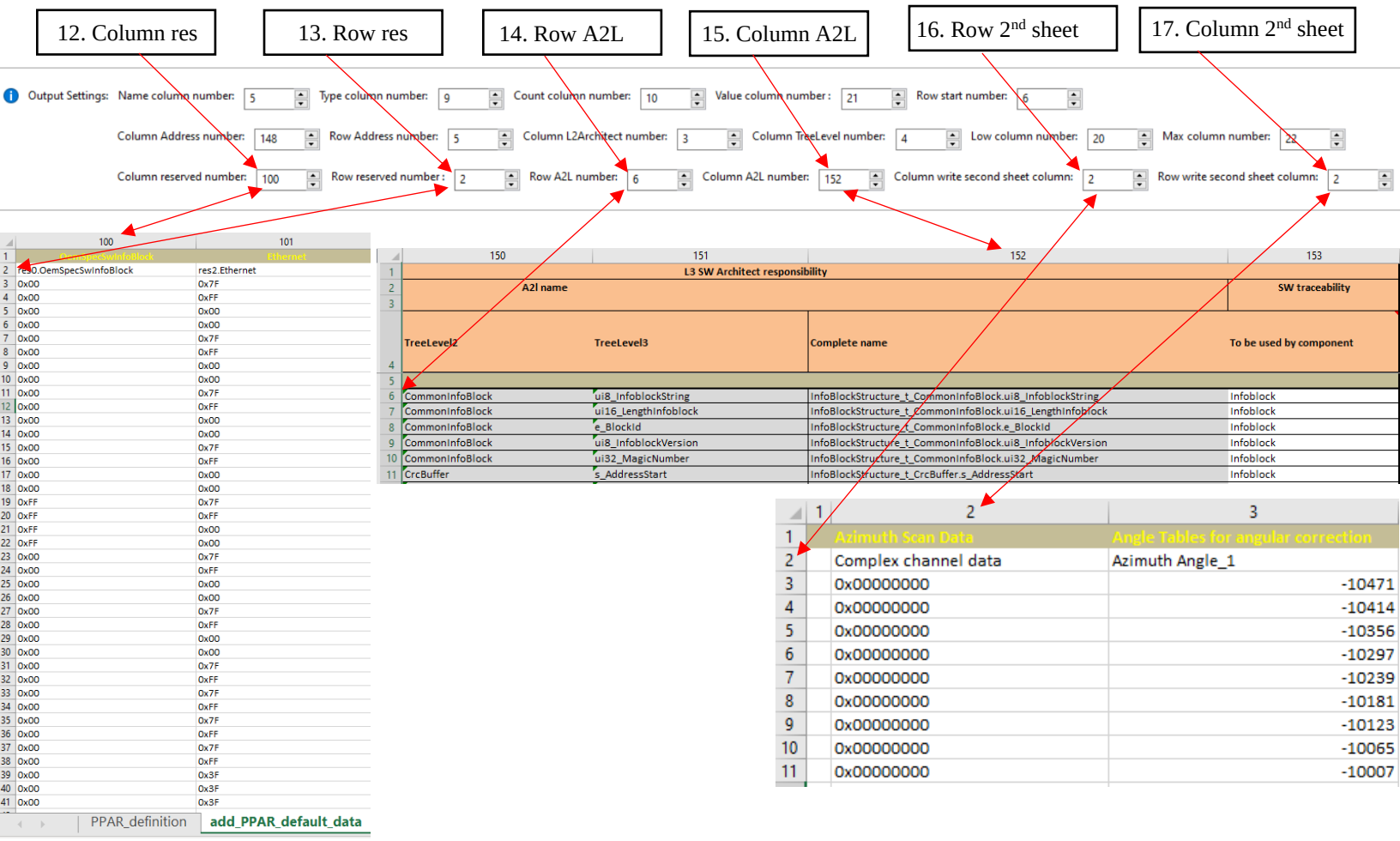
Infoblock	537,001,984		0x20020000										
CommonInfoBlock													
ui8_InfoblockString	537001984	####	0x20020000	UI8	8	1	8	0	GRMSTRM\0'	G R M S T	GRMSTRM\0'
ui16_LengthInfoblock	537001992	####	0x20020008	UI16	1	2	2	0	0x80	0 x 8 0	0x80
e_BlockId	537001994	####	0x2002000A	UI8	1	1	1	0	0x15	0 x 1 5	0x15
ui8_InfoblockVersion	537001995	####	0x2002000B	UI8	1	1	1	0	0x10	0 x 1 0	0x10
ui32_MagicNumber	537001996	####	0x2002000C	UI32	1	4	4	0	0xAA5AA555	0 x A A	0xAA5AA555
CrcBuffer													

10. Set the column number where the program will write the lower limit of the parameter. The program is reading this parameter from the System Excel. The values are written in decimal.
11. Set the column number where the program will write the upper limit of the parameter. The program is reading this parameter from the System Excel. The values are written in decimal.

	1	2	3	4	5	9	10	11	19	20	21	22
1	L2+L3 shared responsiblity											
2	PPAR		generic									
3											Value set 1	
4	Resp.	Status	L2 architect respo	TreeLevel2	a2l name	Type	Count	Total Size	Title	Lower	Default	Upper
122	L3_SW	Agreed	Calibartion	Calibration Val	CalSet1 RX1...4 setting	uint8	4	4		53	0xFF	53
123	L3_SW	Agreed	Calibartion	Calibration Val	align2	uint8	4	4			0x00	
124	L3_SW	Agreed	Calibartion	Calibration Val	CalSet2_AntCompValue	sint16	24	48		-32768	0x7FFF	32767
125	L3_SW	Agreed	Calibartion	Calibration Val	CalSet2 RX-Channel_Calibration	uint16	4	8		10361	0x7FFF	32767
126	L3_SW	Agreed	Calibartion	Calibration Val	CalSet2 TX1...3	uint8	3	3		0	0xFF	63

CalSet1 RX1...4 setting	VGA gain of RX channels. Standard definition: all RX have the same gain	1	HoMon , RHC	1	no scaling	UI8	4	doors://rbgs854a:4 0000/?version=2&pr odID=0&view=0000 0001&urn=urn:telel

Col Title	RX1
lower	0x35
default	0xFF
upper	0x35



12. Set the column number where the program will start writing the reserved space with values bigger than 32 bytes. All reserved values bigger than 32 bytes are written in the second sheet called “add_PPAR_default_data”, these values are written after column 100.
13. Set the row number from where the program will start writing values of the reserved space bigger than 32 bytes.
14. Set the row number from where the program will start writing A2L names.
15. Set the column number where the program will write A2L names.
16. Set the row number from where the program will start writing parameters with values bigger than 32 bytes.
17. Set the column number where the program will start writing the parameters with values bigger than 32 bytes. All values bigger than 32 bytes are written in the second sheet called “add_PPAR_default_data”, these values are written after column 2.

7. Run and Cancel buttons Section

After filling up all the interface, press the Run button and wait for the program to finish execution. If an error occurs during this time, the full description of the error will be written in the log file and the user will have a friendly message description.