

A photograph of an open electrical control cabinet containing six ABB AC500 V3 inverters mounted in a row. Each inverter is white with a digital display and control buttons. The displays show numerical values: 39.95, 1.10, and 45.9. Below the displays are blue 'HVAC' labels and yellow triangular warning symbols with a lightning bolt. The ABB logo is visible at the top of each unit. The cabinet's interior is metallic and shows mounting rails and some wiring at the bottom.

AC500 V3 - Serial ASCII communication

Webinar – best practices



ASCII Code

What does it stand
for?

ASCII Code

What does it stand for?

- **American Standard Code for Information Interchange (ASCII)** introduced in 1963
- 7-bit character encoding that represents control & printable characters
- $2^7 = 128$ characters

ASCII TABLE

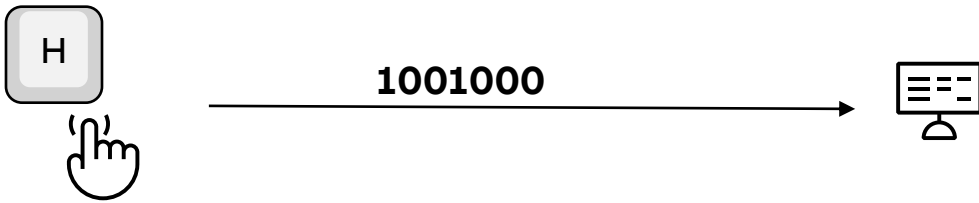
Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[END OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	~
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[					
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135	]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

ASCII Code

What does it stand for?

- **American Standard Code for Information Interchange** (ASCII) introduced in 1963
- 7-bit character encoding that represents control & printable characters
- $2^7 = 128$ characters

$$\begin{aligned} \text{H} &= 72_{10} \\ &= 48_{16} \\ &= 1001000_2 \end{aligned}$$



67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L



AC500 V3 – Serial communication

Working Principle

AC500 V3 – Serial communication

Connection overview

General

- Two wired communication
- UART interface (**U**niversal **A**synchronous **R**eceiver/**T**ransmitter)
 - RS232
 - RS485

AC500 V3

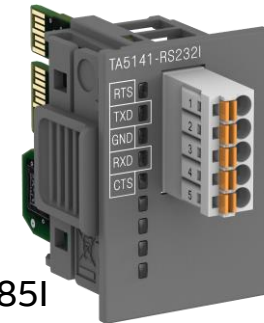


COM1

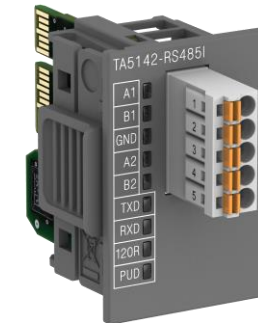
- RS232
- RS485

AC500-eCo V3

TA5141 – RS232I



TA5142 – RS485I

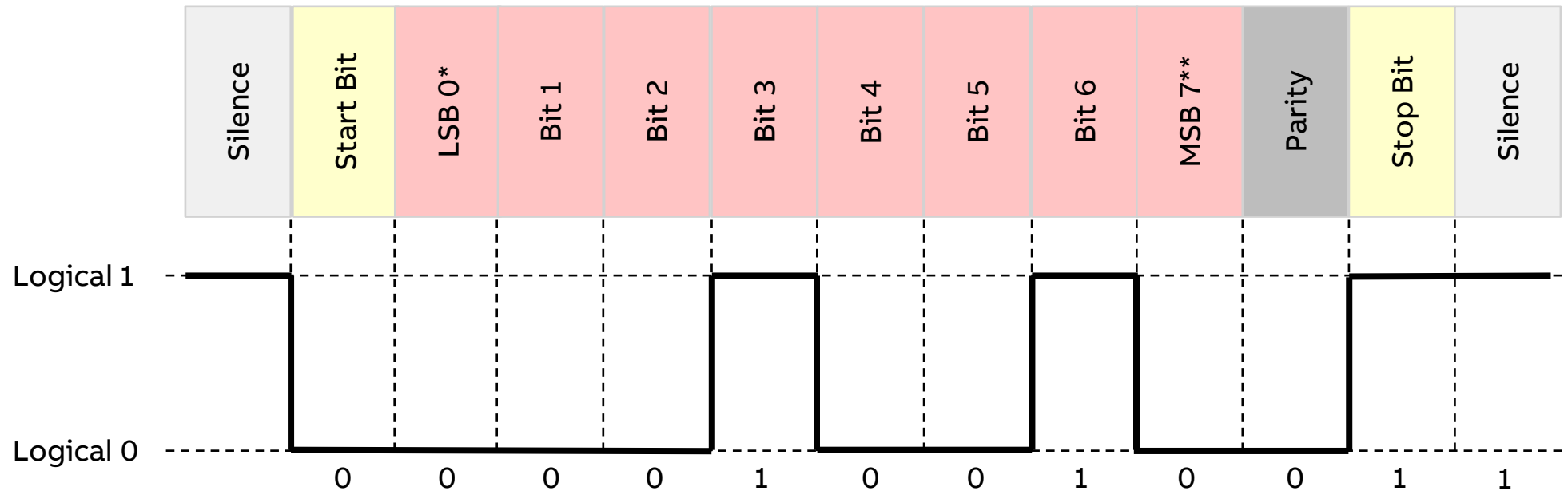


TA5142 – RS485



AC500 V3 – Serial communication

Universal Asynchronous Receiver Transmitter (UART)



$H = 1001000_2$

MSB**

LSB*

* Least significant bit

** Most significant bit

Parameter settings in Automation Builder (AC500-eCo V3)

Devices

- AC500_V3_Serial_ASCII_Communication_AB252
 - PLC_AC500_V3 (PM5650-2ETH - TB5620-2ETH)
 - PLC_AC500_V3_1 (PM5072-T-2ETH)
 - PLC Logic
 - AsciiWrite
 - Diagnosis
 - Enum
 - Function Block's
 - Struct
 - Library Manager
 - PLC_PRG (PRG)
 - Task Configuration
 - Task
 - PLC_PRG
 - VISU_TASK
 - VisuElems.Visu_Prg
 - Visualization Manager
 - MainPage
 - OnBoard_IO (12DI/8DO-T/2DC)
 - IO_Bus
 - Interfaces
 - TA5142_RS485I (TA5142-RS485I)
 - CAA_SerialCom (CAA SerialCom)
 - OptionSlot_2 (<Empty>)
 - OptionSlot_3 (<Empty>)
 - Ethernet
 - OnBoard_RTC

TA5142_RS485I

TA5142-RS485I Parameters

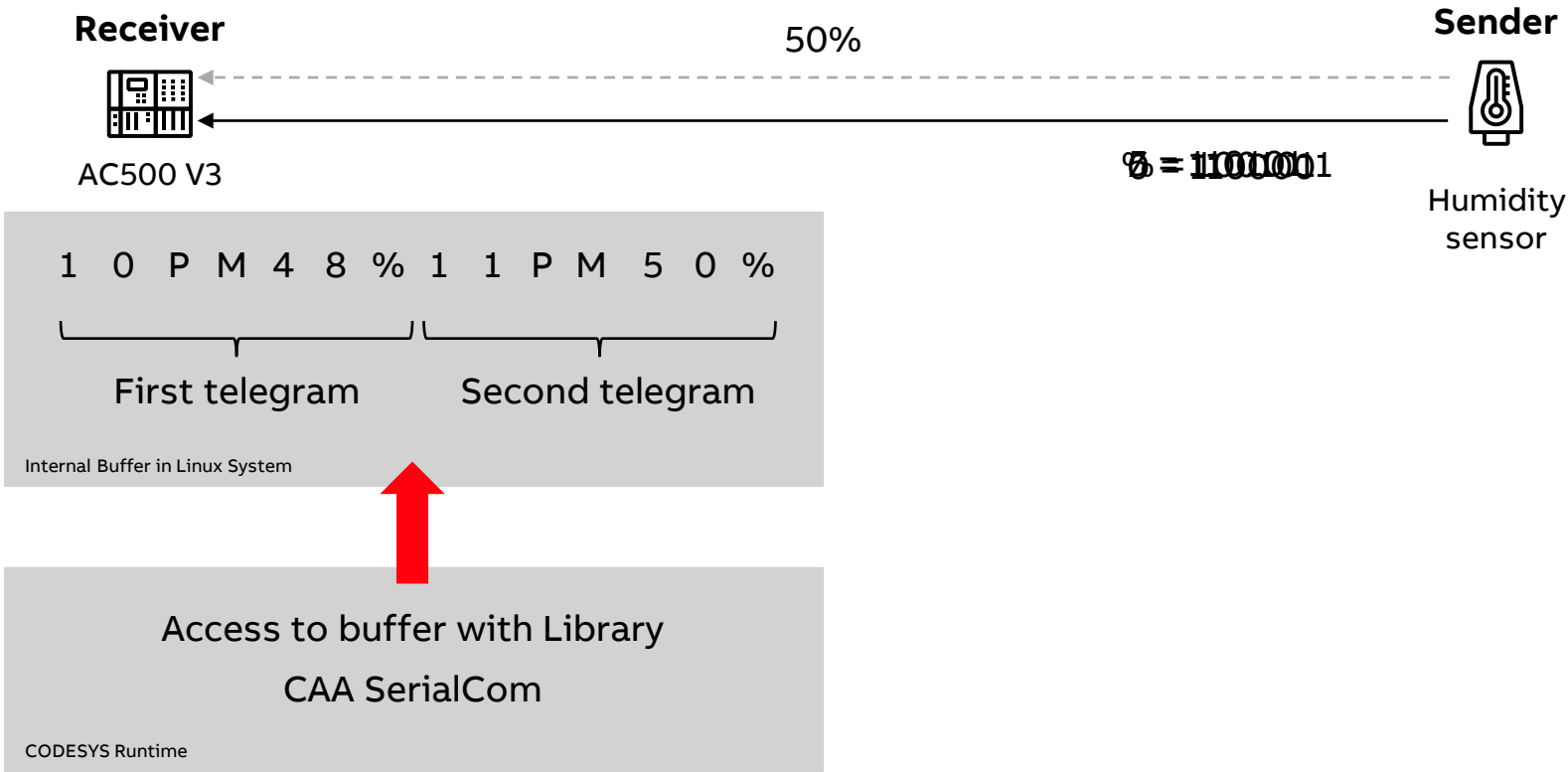
TA5142-RS485I IEC Objects

Information

Parameter	Type	Value	Default Value	Unit
Run on config fault	Enumeration of BYTE	No	No	
Baudrate	Enumeration of DWORD	19200	19200	Bits/s
Parity	Enumeration of BYTE	None	None	
Data Bits	Enumeration of BYTE	8	8	Bits/character
Stop Bits	Enumeration of BYTE	1	1	

AC500 V3 – Serial communication

Sender - Receiver



How do we know which character belongs to which telegram?



ASCII in AC500

Comparison
between V2 and V3

ASCII in AC500

Telegram ending identifier (Online Help)

Telegram ending	Description
None	No telegram ending identifier, i.e., the characters received since last call are provided. The maximum number of characters is limited to 256.
String (check receive)	According to value set for "Telegram ending character", it is checked for 1 or 2 ending characters. The ending character(s) is (are) not passed, i.e., they are not contained in DATA area. Common control characters: Carriage Return (CR) or Line Feed (LF) Example: 1 0 P M 4 8 % \$R\$N (IEC values for CRLF)
Telegram length	Telegram received is reported once the number of characters defined in "Telegram ending value" is received. Example: 1 0 P M 4 8 % = 7 characters
Character timeout (V2) Telegram timeout (V3)	The number of characters set for Telegram ending value and the interface parameters ("Transmission rate", "Parity", "Data bits" and "Stop bits") are used to calculate the silent time. Telegram received is reported if the silent time between two characters is $\geq$ the calculated silent time. In AC500 V3, the character timeout is part of the operating system. The ending identifier is replaced by Telegram timeout and stops reading if the time between two received characters is exceeded.
Duration	Telegram received is reported once the time set for "Telegram ending value" (in [ms]) is elapsed. The time starts with the first FALSE -> TRUE edge at input of the receive function block.

ASCII in AC500

AC500 V2 – Configuration and programming

AC500 V2 – Parameter settings

Parameter	Type	Value	Default Value	Unit	Description
Enable login	Enumeration of BYTE	Disabled	Disabled		Check for CoDeSys login
RTS control	Enumeration of BYTE	None	None		RTS control must be set to 'telegram' for RS485 !
TLS	WORD(0..65535)	0	0		Set the TLS (carrier leading time) in ms (TLS >= CDLY)
CDLY	WORD(0..65535)	0	0		Set the CDLY (carrier lagging time) in ms (CDLY <= TLS)
Character timeout	WORD(0..65535)	0	0		Set the Character timeout in characters must be 0, when telegram ending is character timeout
Telegram ending selection	Enumeration of BYTE	None	None		Telegram ending selection
Telegram ending character	BYTE(0..255)	None	0		Set the telegram ending character
Telegram ending value	WORD(0..65535)	String (check receive) Telegram length Duration Character timeout	0		Set the telegram ending value in ms or characters
Checksum	Enumeration of BYTE		None		Select checksum type
Baudrate	Enumeration of DWORD		19200	Bits/s	Set the baudrate in Bits per seconds
Parity	Enumeration of BYTE	None	None		Set the parity Bit type
Data Bits	Enumeration of BYTE	8	8	Bits/character	Set the character size
Stop Bits	Enumeration of BYTE	1	1		Set the number of stop Bits per character 2 means 1,5 when character size is 5 Bits
Run on config fault	Enumeration of BYTE	No	No		Start PLC program even on configuration fault

Functionblock's to send and receive data

standard.lib
lecsfc.lib 2.6.14 11:37:46
Util.lib 26.9.14 11:12:46
SysLibTime.lib
SysTaskInfo.lib 12.3.23 12:59:16
SysLibMem.lib
SysLibInitLibrary.lib
SYSLIBCALLBACK.LIB
BusDiag.lib
SysInt_AC500_V10.lib 12.3.23 12:59:16
SysExt_AC500_V10.lib
ASCII_AC500_V10.lib 12.3.23 12:59:16
Ethernet_AC500_V10.lib 12.3.23 12:59:16

POUs

COM_REC (FB)
COM_SEND (FB)

FUNCTION_BLOCK COM_REC (*Supported in PLC FW Version V1.0
(*Reads data sets from receiving buffer of serial interface*)

VAR_INPUT
EN: **BOOL** := **FALSE**; (*1: receiving of data by EN = TRUE*)
COM: **BYTE** := 0; (*2: 1->COM1, 2->COM2 *)
DATA: **DWORD** := 0; (*3: address for received data *)
END_VAR

VAR_OUTPUT
DONE: **BOOL** := **FALSE**; (*1: execution finished when output C
ERR: **BOOL** := **FALSE**; (*2: error occurred during execution w
ERNO: **WORD** := 0; (*3: error code*)

COM_REC

— EN : BOOL DONE : BOOL
— COM : BYTE ERR : BOOL
— DATA : DWORD ERNO : WORD
 LEN : WORD

Telegram ending is validated according to Automation Builder setting

ASCII in AC500

AC500 V3 – Configuration and programming

AC500 V3 – Parameter settings

TA5142_RS485I x

TA5142-RS485I Parameters					
Parameter	Type	Value	Default Value	Unit	
Run on config fault	Enumeration of BYTE	No	No		
Baudrate	Enumeration of DWORD	19200	19200	Bits/s	
Parity	Enumeration of BYTE	None	None		
Data Bits	Enumeration of BYTE	8	8	Bits/character	
Stop Bits	Enumeration of BYTE	1	1		

Functionblock's to send and receive data

Library Manager x

Add Library X Delete Library Properties Details Placeholders Library Repository Icon Legend... Summary...

Libraries used in application 'PLC_AC500_V3.AsciRead'

Name	Namespace	Effective Version
3SLicense = 3SLicense, 3.5.18.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.18.0
AC500_Com = Com, 1.0.2.3 (ABB)	AC500_Com	1.0.2.3
AC500_ComBase = ComBase, 1.0.2.1 (ABB)	AC500_ComBase	1.0.2.1
AC500_StateMachine = StateMachine, 1.0.6.6 (ABB)	AC500_StateMachine	1.0.6.6
CmpErrors2 Interfaces, * (System)	CmpErrors	3.5.19.0
Serial Communication, * (3S - Smart Software Solutions GmbH)	Serial_Communication	3.5.17.0
CAA SerialCom = CAA SerialCom, 3.5.17.0 (CAA Technical Workgroup)	COM	3.5.17.0
CAA Types = CAA Types Extern, 3.5.17.0 (CAA Technical Workgroup)	CAA	3.5.17.0
SysTypes2 Interfaces, * (System)	SysTypes	3.5.17.0
AC500_DiaqCpu = DiaqCpu, 1.2.2.1 (ABB)	AC500_DiaqCpu	1.2.2.1

Contents of selected library 'CAA SerialCom, 3.5.17.0 (CAA Technical Workgroup)'

CAA SerialCom

Enums

Function Blocks

Close

Open

Read

Write

Global Variables

Structs

Library Information

Details about selected library element 'Overview'

Documentation

CAA SerialCom Library Documentation

Company: CAA Technical Workgroup

Title: CAA SerialCom

Version: 3.5.17.0

Categories: Intern|CAA|System

Namespace: COM

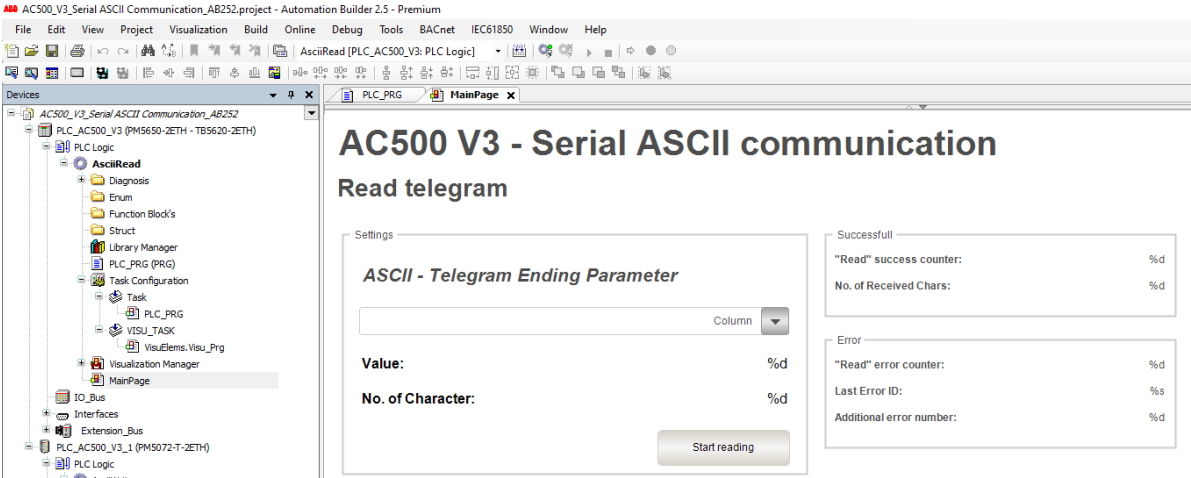
Author: 3S - Smart Software Solutions GmbH

Placeholder: CAA SerialCom

Telegram ending cannot be set via Automation Builder

ASCII in AC500

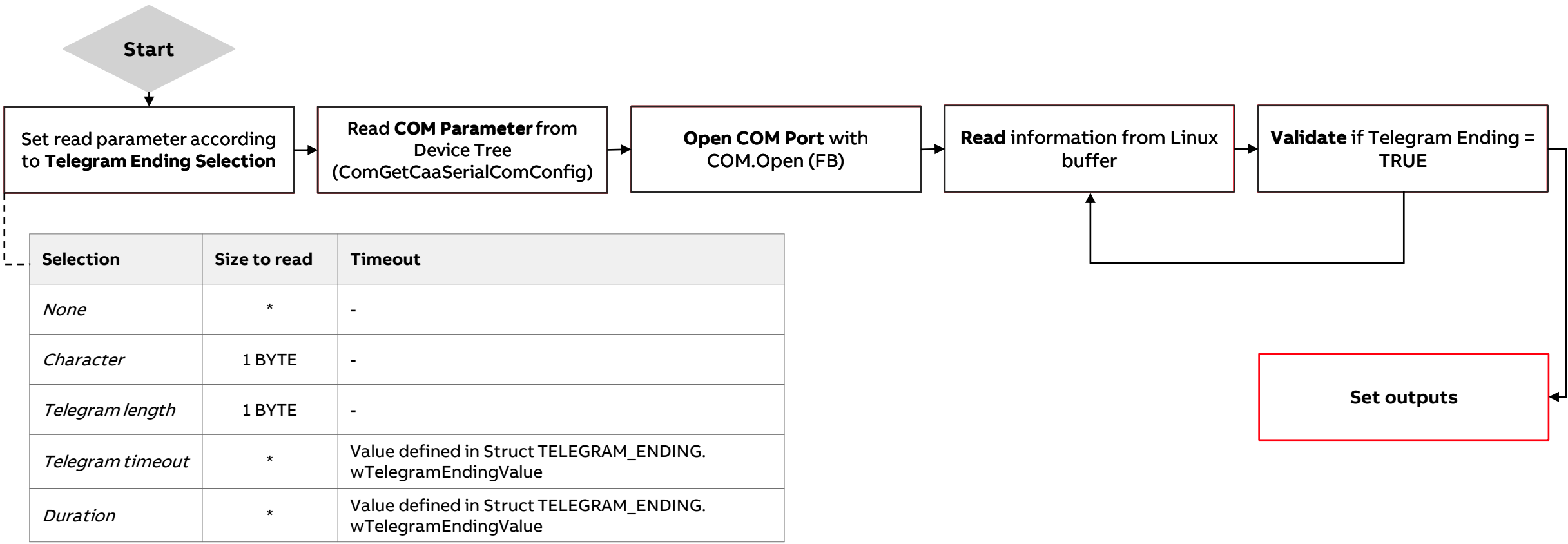
Application Example – AC500 V3 Serial ASCII Communication



[Link](#) to Application Example in ABB Library

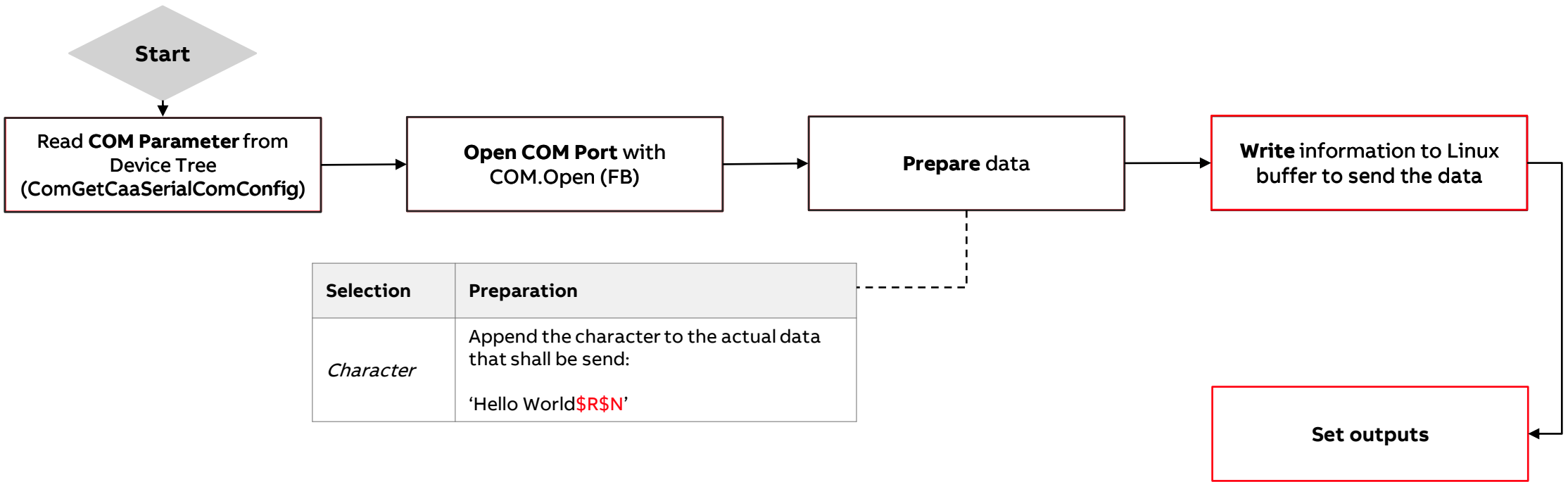
ASCII in AC500

Integration of ASCII Read in AC500 V3



ASCII in AC500

Integration of ASCII Write in AC500 V3





Live demonstration



Q&A



ABB