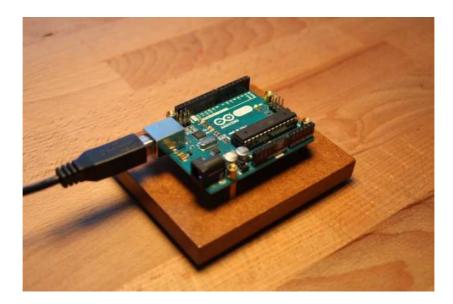
Getting started with WawiLib using the Arduino programming port

1.	Intr	oduction	2
	1.1.	Objective of this document	2
	1.2.	Software and hardware requirements	2
	1.3.	Required user experience	2
2.	Inst	tall WawiLib Software	3
3.	Loa	d the Arduino board with the demo sketch	5
4.	Wa	wiLib user interface overview	6
5.	Wa	wiLib communication link setup	10
6.	Rea	ad and write variables with WawiLib	14
	6.1.	Watch variables	14
	6.2.	Modify variables	14
7.	Rec	ord variable to file (introduction)	16
8.	Rec	cord .print() output to file (introduction)	20
9.	Intr	oduction to WawiLib breakpoints	25
10	. F	urther reading	27



1. Introduction

1.1. Objective of this document

The objective of this demo is to describe step by step how to get WawiLib up and running with a very small Arduino example program (sketch).

Many users know the Arduino "Blink" sketch. "Blink" is designed to blink an on-board LED as you can find on many Arduino boards. In this document, you will learn how to create "WawiBlink" – the WawiLib version of "Blink".

"WawiBlink" blinks the same LED, but with variable time intervals. The time the LED is on and the time it is off is defined by 2 variables: *delayOn* and *delayOff*. The number of blinks is stored in the variable *blinkCounter*.

In this demo, you will learn how to monitor and modify *delayOn, delayOff* and *blinkCounter* while the sketch is running on the Arduino board. The demo will also demonstrate how you can record the value of *blinkCounter* in an .xml, .xlsx or .csv file that can later be opened in Microsoft Excel, LibreOffice or a program you have written yourself.

You will also learn how to create diagnostics messages that will be displayed in the console output window of the WawiLib-PC application. The example uses this function to report the state changes of the onboard LED.

This demo also shows how to record the output of sketch .print() statements in a disk file on your PC and the option to use breakpoints in your sketch.

1.2. Software and hardware requirements

The Arduino IDE (in this example 1.8.15) and WawiLib V2.0.x both need to be installed on your PC. The demo runs with licensed and unlicensed versions of WawiLib. During the grace period of 2 months, you can test and use all functions without registration. After this period registration is required in order to access all functions. At this time registration is free. In the future a small contribution might be required to register in order to support the website.

WawiLib supports multiple interface types: serial, software serial, USB, USB-native, TCP/IP, UDP/IP via cable or Wi-Fi. In this demo, the USB programming port of the Arduino board is used.

The hardware you need is an Arduino board, a USB programming cable and a Windows PC (32 or 64 bit). The requirements are the same as those to run "Blink". In this demo, we will use the Arduino UNO board but other boards can be used in a similar or even identical way.

For the demo, the only difference between the UNO and other boards is the IO location of the LED. Samples for other boards are provided in the "Examples" section of the Arduino IDE after installing WawiLib. You can modify the definition of the constant "LED" yourself in the sample sketches if there is no sample for your board provided with WawiLib.

1.3. Required user experience

This demo assumes that you know how to edit, compile and download Arduino programs. You should also have basic computer skills such as downloading and installing Windows programs.

2. Install WawiLib Software

This section describes the steps you have to follow in order to install the WawiLib program and the WawiSerialUsb Arduino library. If both have been correctly installed on your PC, you can skip this section.

- ✓ Download the WawiLib installer from <u>www.sylvestersolutions.com</u>.
- ✓ Install WawiLib using the downloaded WawiLib32.msi or WawiLib64.msi installer.
- ✓ Start WawiLib.
- WawiLib will unpack the zipped WawiLib Arduino libraries and put them in the library directory of the Arduino IDE.
- ✓ Open the Arduino IDE.
- ✓ Check the presence of the installed libraries:

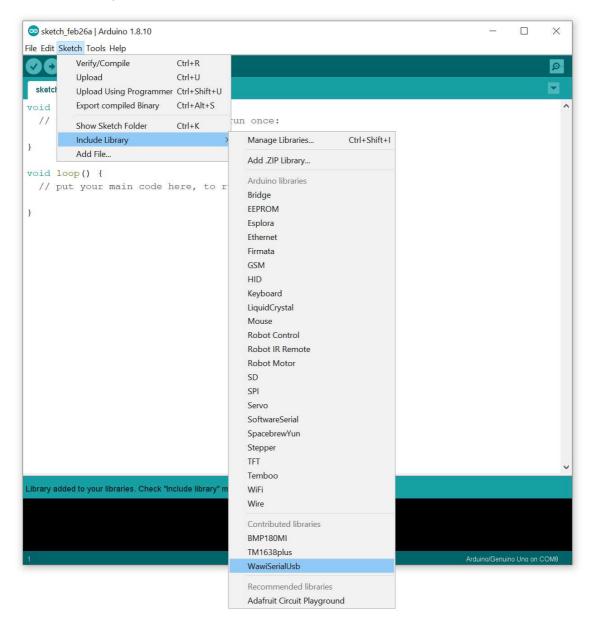


Fig. 2.1. Check the installation of the WawiLibSerialUsb library in the Arduino IDE.

The libraries WawiSerialUsb, WawiEthernet and WawiWifi can be found in: C:\Users\[your user name]\Documents\Arduino\libraries.

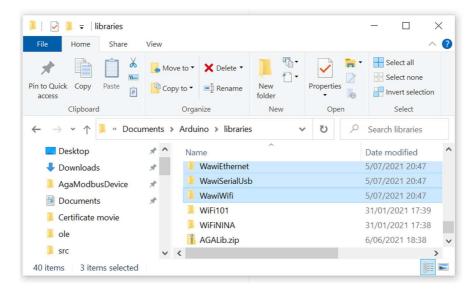


Fig. 2.2. Unpacked Libraries after installing WawiLib.

Note: 1) if, by exception, automatic installation of the libraries fails, you can manually unzip the WawiSererialUsb.zip, WawiEthernet.zip and WawiWifi.zip in the Documents\Arduino\Libraries directory. The libraries can be found in the installation directory of WawiLib.exe itself.

Note: 2) Manual installation of libraries can be triggered in the WawiLib menu "Settings\Preferences and license". In tab "WawiLib Arduino libraries" press the button "Install\Update WawiLib Libraries for Arduino".

Layout	License	Startup and exit	Comm. timing	Wawilib arduino libraries	Window refresh	
Wawi	Lib Arduin	no compressed libr	ary location:			
C:\V	cProjects2	019\WawiLib				
Instal	lation prog	gress:				
Pres	s install/up	odate to unzip libra	ries to update Ar	duino libraries directory.		^
				VcProjects2019\WawiLib\W	VawiSerialUsb.zip:	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	les\WawiBlinkDeb	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	les\WawiBlinkDue	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	oles\WawiBlinkEsp8	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	les\WawiBlinkMeg	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	oles\WawiBlinkMKF	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	les\WawiBlinkMKF	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	les\WawiBlinkNan	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	les\WawiBlinkNan	
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp	les\WawiBlinkNan	
11.4	ating CALL					
Upa	aling Civo	sers\Johi\Docume	nts\Arduino\libra	aries\WawiSerialUsb\examp	oles\WawiBlinkNod	
				aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp		
Upd	ating C:\U	sers\Johi\Docume	nts\Arduino\libra		oles\WawiBlinkSeria	
Upd Upd	ating C:\U ating C:\U	sers\Johi\Docume sers\Johi\Docume	nts\Arduino\libra nts\Arduino\libra	aries\WawiSerialUsb\examp	oles\WawiBlinkSeria oles\WawiBlinkSeria	
Upd Upd Upd	ating C:\U ating C:\U ating C:\U	sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume	nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra	aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp	oles\WawiBlinkSeria oles\WawiBlinkSeria oles\WawiBlinkSoft	
Upd Upd Upd Upd	ating C:\U ating C:\U ating C:\U ating C:\U	sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume	nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra	aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp	oles\WawiBlinkSeria oles\WawiBlinkSeria oles\WawiBlinkSoft oles\WawiBlinkUno	
Upd Upd Upd Upd	ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U	sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume	nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra	aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp	oles\WawiBlinkSeria oles\WawiBlinkSeria oles\WawiBlinkSoft oles\WawiBlinkUno oles\WawiBlinkUsbl	
Upd Upd Upd Upd Upd Upd Upd	ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U	sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume sers\Johi\Docume	nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra	aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp	oles\WawiBlinkSeria oles\WawiBlinkSeria oles\WawiBlinkSoft oles\WawiBlinkUno oles\WawiBlinkUsbl oles\WawiBlinkUsbl oles\WawiBlinkUsb'	
Upd Upd Upd Upd Upd Upd Upd Upd	ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U	sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei	nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra	aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp aries/WawiSerialUsb/examp	oles\WawiBlinkSeria oles\WawiBlinkSoft oles\WawiBlinkUno oles\WawiBlinkUsbl oles\WawiBlinkUsbl oles\WawiBlinkUsb' oles\WawiEmpty\W oles\WawiRecUsb\	
Upd Upd Upd Upd Upd Upd Upd Upd	ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U ating C:\U	sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei sers\Johi\Documei	nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra nts\Arduino\libra	aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp aries\WawiSerialUsb\examp	oles\WawiBlinkSeria oles\WawiBlinkSoft oles\WawiBlinkUno oles\WawiBlinkUsbl oles\WawiBlinkUsbl oles\WawiBlinkUsb' oles\WawiEmpty\W oles\WawiRecUsb\	*

Fig. 2.3. Manual install of Arduino libraries.

3. Load the Arduino board with the demo sketch

Many of the Arduino libraries come with examples. WawiLib is not an exception. In this demo, we will use the sketch called WawiBlinkUsb.ino.

- ✓ Go to File\Examples\WawiSerialUsb\WawiBlinklUsb in the Arduino IDE.
- ✓ Open and compile WawiBlinkUsb and download the sketch to your Arduino board.
- ✓ Check if the program was properly downloaded by looking at the LED on the board. The LED should blink 500ms on and 500ms off.

```
#include<WawiSerialUsb.h>
WawiSerialUsb WawiSrv;
#defineLED 13
// test variables for demo:
int delayOn = 500;
int delayOff = 500;
int blinkCounter = 0;
// make variables of interest known to WawiLib:
// thisfunction is used in WawiSrv.begin(....)
void wawiVarDef()
{
    WawiSrv.wawiVar(delayOn);
    WawiSrv.wawiVar(delayOff);
    WawiSrv.wawiVar(blinkCounter);
}
void setup()
{
    Serial.begin(115200);
    WawiSrv.begin(wawiVarDef, Serial, "My Arduino");
    pinMode(LED, OUTPUT);
}
void loop()
{
    blinkCounter++;
    WawiSrv.print("WawiSrv.Print() demo in loop() function, blinkcounter = ");
    WawiSrv.println(blinkCounter);
    WawiSrv.println("LED is ON.");
    digitalWrite(LED, HIGH);
    WawiSrv.delay(delayOn);
    WawiSrv.println("LED is OFF.");
    digitalWrite(LED, LOW);
    WawiSrv.delay(delayOff);
    WawiSrv.loop();
```

Fig. 3.1. Minimal Arduino example WawiBlinkSerial.ino

4. WawiLib user interface overview

✓ Start WawiLib on your PC:

en e	E Open	■ Save	🖶 Print	ि Сору	Ж Cut	🗈 Paste	0ffline	C Setup()	To Loop(↓ Write all	► Continue	e brkpt) brkpt	
Avai	lable variable	es		Interf	ace/Ard.	ID Va	riable name	Actual	value	Write value	Format	Recorde	ज Brkp on L	Varia '
			1										Sinds out	
			2											
			3											
			4											
			5											
		_	6											
		4												>
Index	Time			Node			Message							1
000	7/07/2021			Welcome			Welcome to	o WawiLil	o V2.0.1					
001	7/07/2021	21:31:46	.891											
002	7/07/2021	21:31:47	.043	License de	tails		License typ	e: D = Pr	emium;	valid until 202	21-8-14			
003	7/07/2021	21:31:47	.046	License de	tails		License ow	ner: john.	do@ma	ail.com				
004	7/07/2021	21:31:47	.048	License de	tails		License seri	al numbe	er:					
005	7/07/2021	21:31:47	.051	License de	tails		-> #variabl	es = unlir	nited					- 1
006	7/07/2021	21:31:47	.054	License de	tails		-> serial co	mmunica	ition = y	/es				
007	7/07/2021	21:31:47	.056	License de	tails		-> TCP/IP-U	JDP-WiFi	commu	unication = ye	s			
800	7/07/2021	21:31:47	.059	License de	tails		-> #record	ed variab	les = un	limited				
009	7/07/2021	21:31:47	.062	License de	tails		-> dll link =	no no						
010	7/07/2021	21.21.47	OGE											~

Fig. 4.1. WawiLib startup screen

The main window is split into 3 parts. The upper part contains a grid and a tree control, the bottom part contains a list box.

Once connected to the Arduino, the tree control shows all shared (static) variables in your sketch. In the grid control you enter the variables of your interest, the interface to be used, some parameters related to the variable itself and the data recorder(s) to be used. Drag & drop from the tree to the grid are also possible.

Interface and recorders can be configured using the "Settings" menu.

✓ Right click on the grid in the top window for additional options:

🔊 New	Dpen	₽ Save	👼 Print	Copy	X Cut P	🗈 aste	0ffline	Setup()	Loop		↓ ite all	► Continue	• brkpt) brkpt		
Avai	lable variable	s		Interfa	ce/Ard. ID	Var	iable name	Actual	value	Write	value	Format	Recorder		Va	ria '
			1			N	DI 1 (1					
			2			13	Display fo	rmats		>						
			3			~	Offline									
		_	4				Setup()				L					
		_	5				Loop()									
			6				Write row		C	trl+W	L					
		<					Write sele	cted field	s						1	>
Index	Time			Node			Write all		1	Alt+W						
000	7/07/2021	21:31:46.	.887	Welcome			Cut		(Ctrl+X						
001	7/07/2021 2						Сору		(Ctrl+C						
002	7/07/2021			License det			Paste			Ctrl+V	il 202	21-8-14				
203	7/07/2021 2			License det						curre						
004	7/07/2021			License det			Clear row(
005	7/07/2021 2			License det			Insert row									1
006	7/07/2021 2			License det			Insert row	s								
007	7/07/2021 2	21:31:47.	.056	License det	ails		Delete rov	/(s)			= ye	IS				
800	7/07/2021 2			License det			Clear entir	e table								
009	7/07/2021 2			License det	ails		Variable p	operties								
010 <	7/07/2021	21.31.47	065				vanable p	operues			-				,	

Fig. 4.2. WawiLib grid options.

Most of the options do not require additional comment, but the sub option "Display format" allows you to select various display formats for the variables in the grid.

The lower part is an output window used to report what WawiLib is doing. It is very handy if you have trouble going online on your board or if you want to see if a variable change was written to your Arduino board properly.

✓ Right click on the bottom window (output window) for additional options:

没 Wa	wiLib-PC [C:	\Users\J	ohi∖Docum	ents\NoNa	me.Wif]-[C:\Users\Jc	hi\Docume	nts\NoN	ame.Wv	/f]			_		×
File Ec	lit Settings	Help													
1		-	8	1	ж	<u>î</u>	T	1	6	÷	₽	•			
New	Open	Save	Print	Сору	Cut	Paste	Offline	Setup()	Loop(0 Write a	ll Continu	e brkpt	brkpt		
<mark>Avail</mark>	lable variable	es		Interf	ace/Ard.	ID Vari	able name	Actual	value	Write valu	e Format	Recorder			Varia ^
			1												
			2												
			3												
			4												
			5												
			6												~
			<												>
Index	Time			Node			Message								^
000	7/07/2021	21:31:46	5.887	Welco	D: 1		\\/_lA	- \&/!! :!	- 1/2 0 1						
001	7/07/2021	21:31:46	5.891	1 Jer	Display	.print() me	ssages								
002	7/07/2021	21:31:47	7.043	Licens 🖌	Display	diagnostic	s messages				021- 8-1 <mark>4</mark>				
003	7/07/2021			Licens	Display	communic	ation proto	col mess	ages						
004	7/07/2021			Licens	Display	data record	dina								
005	7/07/2021			Licens											
006	7/07/2021			Licens	Display	output wir	idow record	ling							
007	7/07/2021			Licens	Copy se	elected text				Ctrl+C	/es				
800	7/07/2021			Licens	Clear W	indow									
009	7/07/2021			Licens 🗸	Automa	itic scroll									~
010 《	7/07/2021	21:31:47	065		Reset vie	ew									>
						Of <mark>fli</mark> n	e Autow	rite on	No da	ita recorders	No output	recorders	No in	terfaces	active

Fig. 4.3. WawiLib output window options.

In the figure above, you see the popup menu where you can enable and disable different tracing settings.

- Display .print() messages: display the output of WawiSerialUsb.print() messages used in your sketch for diagnostics and other purposes.
- Display diagnostics messages: display the output of general WawiLib diagnostics messages.
- Display communication protocol messages: display the communication messages as they are exchanged between the PC and the Arduino board.
- Display data recording: display the data written to disk by the data recorders (log variables).
- Display output recording: display the data written to disk by the output recorders (log .println() output.)
- Automatic scroll: If activated, WawiLib will automatically scroll to the latest message in the output window every time a new message arrives.

The image above gives an incomplete overview of the various fields. Therefor I will use a more extended case for the bottom status line. This is the output of the WawiDemoControllinoTcpCable

demo also included with WawiLib. The demo uses an Ethernet TCP interface on an Controllino Arduino Mega 2560 compatible PLC with generic WS5100/5500 Ethernet connection.

P New	Dpen Save Print	Copy	K 🗈 Taste Offli		ta v oop() Write al	► Continue		⊠ rkpt	
	p1\MyControllino		Interface/Ard. ID	Variable name	Actual value	Write value	Format	Recorder	Variable address and status
1 mar	Single variable	1	tcp1/MyControllino	wawiTimer0	460		INT		@wawiTimer0=0x056C [8 byte] x 1 VAR READING OK
	blinkCounter	2	tcp1/MyControllino	wawiTimer0Max	500	500	INT	REC1	@wawiTimer0Max=0x0224 [2 byte] x 1 VAR_READING
	H wawiTimer0	3	tcp1/MyControllino	wawiTimer1	1157		INT		@wawiTimer1=0x055E [8 byte] x 1 VAR_READING_OK
	# wawiTimer0Max	4	tcp1/MyControllino	wawiTimer1Max	2000	2000	INT		@wawiTimer1Max=0x0222 [2 byte] x 1 VAR READING
	wawiTimer1	5							
	H wawiTimer1Max	6	tcp1/MyControllino	cycleCounter	614		INT		@cycleCounter=0x0413 [2 byte] x 1 VAR_READING_O
	Array	7							
	message[029]	8	tcp1/MyControllino	controllino_d[0]	1		INT		@controllino_d=0x0415 [1 byte] x 5 VAR_READING_OI
	⊟ controllino d[04]	9							
	[1 byte] x 5	10	tcp1/MyControllino	controllino_d[1]	0		INT		@controllino_d=0x0415 [1 byte] x 5 VAR_READING_OI
	controllino_d[0]	11	tcp1/MyControllino	controllino_d[2]	0		INT		@controllino_d=0x0415 [1 byte] x 5 VAR_READING_OI
	-controllino_d[1]	12	tcp1/MyControllino	controllino_d[3]	1		INT		@controllino_d=0x0415 [1 byte] x 5 VAR_READING_OI
	-controllino_d[2]	13	tcp1/MyControllino	controllino_d[4]	0		INT		@controllino_d=0x0415 [1 byte] x 5 VAR_READING_OI
	-controllino_d[3]	14							
	controllino_d[4]	15	tcp1/MyControllino	message[029]	Hello world		STRING		@message=0x0200 [1 byte] x 30 VAR_READING_OK - 💙
	controllino_relay[04]	<							>
ndex	Time	Node		Me	ssage				^
13	10/07/2021 17:56:52.460	tcp1/192.16	8.1.190-192.168.1.88/My	Controllino @c	ontrollino_ai = 0	x3fe [1 byte] x	10		
14	10/07/2021 17:56:52.469	tcp1/192.16	8.1.190-192.168.1.88/My	Controllino @v	vawiTimer0 = 0x	56c [8 byte] x	1		Display .print() messages
15	10/07/2021 17:56:52.478		8.1.190-192.168.1.88/My				-		 Display diagnostics messages
16	10/07/2021 17:56:52.487	Cardbook State Party	8.1.190-192.168.1.88/My		vawiTimer1 = 0x				Display communication protocol messages
17	10/07/2021 17:56:52.496	and the state	8.1.190-192.168.1.88/My				2		
18	10/07/2021 17:56:52.670	REC1			ening file C:\User			ataRecorded	.cs 🗹 Display data recording
19	10/07/2021 17:56:52.670	REC1			te header: date ti		100		Display output window recording
20	10/07/2021 17:56:59.100		8.1.190-192.168.1.88/My						Copy selected text Ctrl+
21	10/07/2021 17:57:00.061	REC1			(o.t.): 10/07/202				Class Window
22	10/07/2021 17:57:04.914	tcp1/192.16	8.1.190-192.168.1.88/My	Controllino [Ro	w 2] Writing vari	able wawiTime	er0Max=500	(INT) comple	ete
23	10/07/2021 17:57:06.061	tcp1/192.16	8.1.190-192.168.1.88/My	/Controllino [Ro	w 4] Writing vari	able wawiTime	er1Max=200	0 (INT) comp	let 🖌 Automatic scroll
24	10/07/2021 17:57:07.100	tcp1/192.16	8.1.190-192.168.1.88/My			and the second second second			Reset view
25	10/07/2021 17:57:10.032	REC1			(o.t.): 10/07/202				
26	10/07/2021 17:57:15.100	tcp1/192.16	8.1.190-192.168.1.88/My	Controllino The	e running light co	mpleted cycle	nr: 613		

Fig. 4.4. WawiLib overview in a Controllino TCP configuration.

At the bottom of the WawiLib window there is a status line indicating the statuses of the application. The line is subdivided in different fields. I will describe the various fields using the example as displayed:

- "Loop()": the target status of the communication interfaces {"Offline", "Setup", "Loop" } Setup()=Arduino is executing setup function, Loop()=Arduino is executing Loop() function. Note: Variable exchange is only available in Loop() mode, .print() is available in Setup() and in Loop() modes.
- "Autowrite on": status Autowrite (See above; "ENTER" key triggers a variable write command for the line in the grid with the selected cell.)
- "REC1 [RECO_WAIT_TRIG] cnt=2": the status name of the recorder named REC1, its FSM (finite state machine) status (=no tags selected for recording). The actual number of data records written to disk or memory (memory for excel .XLM file format) is 2.
- "No output recorders": WawiLib can record .print() output from the sketch into an output file. In this case no recorders for this kind of data are defined.
- "TCP=MyControllino=192.168.1.190-192.168.1.88/49152 [ITF_LOOP]": An interface of type TCP is active. The library was initialized (WawiSrv.begin() function) with parameter value "MyControllino" for the name of the board. The interface card on the PC has IP 192.168.1.168 and the Arduino board has IP 192.168.1.88. TCP port 49152 is used on the Arduino/Controllino side. The actual status of the communication FSM (finite state machine) is ITF_LOOP.

• "Msg.ok/.tot 708/708: There are 708 data telegrams exchanged OK between the Arduino on a total of 708 telegrams.

WawiLib supports multiple interfaces of multiple boards and multiple data recorders at the same time. Therefor the fields "TCP1[...]" and "REC1[...]" display the various recorders and various interfaces one after the other in an alternating way.

5. WawiLib Serial-USB communication link setup

One of the biggest challenges going online on a board is finding the right port and the right settings. With this purpose in mind, WawiLib has a wizard to scan serial/USB ports with various settings to check for the presence of one (or multiple) Arduino board(s).

enal communicati	on scan settings E	thernet UDP or Tcp/I	p commu	nication scan sett	ings		
Baudrate 300 2400 9600 ✓ 38400 ✓ 115200 250000 1000000	 ☐ 1200 ☐ 4800 ☑ 19200 ☑ 57600 ☐ 230400 ☐ 500000 ☐ 2000000 	Parity Even Odd None Stop bits 1 one 2 two			oard familiy No family (usb to serial,) AVR (UNO, MEGA,) SAM (DUE) SAMD (MKR, ZERO,) MBED (Nano 33 BLE,) megaAVR (UNO WiFi Rev2,) ESP8266 (NodeMcu,)	Serial ports COM1 COM3 COM4 COM26	
can list + scan sta [V]=Active/Alias		Parameters	Board ID	Status scanning	Interface status		Add Update Remove Clear list

✓ Select the WawiLib menu Settings\Communication interfaces:

Fig. 5.1. Serial communication setup.

- ✓ Select the baud rates, board type(s) and serial port(s) that you want to check.
- ✓ Press the button "Add":

erial communca	tion scan settings E	thernet UDP or Tcp/	lp commu	nication scan sett	ings		
Baudrate 300 2400 9600 ✓ 38400 ✓ 115200 250000 1000000	 1200 4800 19200 57600 230400 500000 2000000 	Parity Even Odd None Stop bits 1 one 2 two			oard familiy No family (usb to serial,) AVR (UNO, MEGA,) SAM (DUE) SAMD (MKR, ZERO,) MBED (Nano 33 BLE,) megaAVR (UNO WiFi Rev2,) ESP8266 (NodeMcu,)	Serial ports	
an list + scan s	tatuc						
can list + scan s IVI=Active/Alias		Parameters	Board ID	Status scanning	Interface status		Add
		Parameters 115200,8,N,1,AVR		Status scanning SCAN TODO	Interface status ITF OFFLINE		Add
V]=Active/Alias Øser1	Interface		?	5			Add Update
V]=Active/Alias Øser1 Øser1	Interface T Serial: COM4	115200,8,N,1,AVR	? ?	SCAN_TODO	ITF_OFFLINE		
V]=Active/Alias Øser1 Øser1 Øser1	Interface Serial: COM4 Serial: COM4	115200,8,N,1,AVR 57600,8,N,1,AVR	? ? ?	SCAN_TODO SCAN_TODO	ITF_OFFLINE ITF_OFFLINE		Update Remove
V]=Active/Alias Øser1 Øser1 Øser1 Øser1	Interface Serial: COM4 Serial: COM4 Serial: COM4 Serial: COM4	115200,8,N,1,AVR 57600,8,N,1,AVR 38400,8,N,1,AVR 19200,8,N,1,AVR	? ? ? ?	SCAN_TODO SCAN_TODO SCAN_TODO	ITF_OFFLINE ITF_OFFLINE ITF_OFFLINE		Update
V]=Active/Alias	 Interface Serial: COM4 Serial: COM4 Serial: COM4 Serial: COM4 Serial: COM4 	115200,8,N,1,AVR 57600,8,N,1,AVR 38400,8,N,1,AVR 19200,8,N,1,AVR	? ? ? ?	SCAN_TODO SCAN_TODO SCAN_TODO SCAN_TODO	ITF_OFFLINE ITF_OFFLINE ITF_OFFLINE ITF_OFFLINE		Update Remove
V]=Active/Alias Øser1 Øser1 Øser1 Øser1 Øser1 Øser2	Interface Serial: COM4 Serial: COM4 Serial: COM4 Serial: COM4 Serial: COM4 COM4 Serial: COM4	115200,8,N,1,AVR 57600,8,N,1,AVR 38400,8,N,1,AVR 19200,8,N,1,AVR 115200,8,N,1,AVR	? ? ? ? ?	SCAN_TODO SCAN_TODO SCAN_TODO SCAN_TODO SCAN_TODO	ITF_OFFLINE ITF_OFFLINE ITF_OFFLINE ITF_OFFLINE ITF_OFFLINE		Update Remove

Fig. 5.2. Serial communication setup: scanning/setting up the connection.

- ⇒ You will see all combinations listed up in the "Scan list + scan status" table.
- ✓ Press the button "Start scan".

WawiLib will try all setting combinations in the table one by one. When the scan is completed successfully, one (or more) icon(s) in the scan list will turn green. You can follow the process in the output window of WawiLib and in the "Scan list + scan status" window.

rial communca	tion scan settings E	thernet UDP or Tcp/I	lp communicati	on scan settings			
Baudrate 300 2400 9600 ✓ 38400 ✓ 115200 250000	 1200 4800 ✓ 19200 ✓ 57600 230400 500000 	Parity Even Odd None Stop bits 1 one 2 two		Board familiy Do family (u AVR (UNO, N SAM (DUE) SAMD (MKR MBED (Nanc megaAVR (U	MEGA,) 8, ZERO,)	Serial ports COM1 COM3 COM4 COM26	
	2000000			ESP8266 (No	odeMcu,)		
1000000	tatus		Board ID			itus	Add
an list + scan s V]=Active/Alias]ser1	tatus	Parameters 115200,8,N,1,AVR 57600,8,N,1,AVR	?	Status scanning SCAN_ERR_PARAMETERS SCAN_ERR_PARAMETERS	Interface sta S ITF_READ_S	atus ETTINGS_CHECK_INTERF, ETTINGS_CHECK_INTERF,	Add Update
an list + scan s V]=Active/Alias]ser1]ser1]ser1	tatus Interface To Serial: COM4 To Serial: COM4 To Serial: COM4	Parameters 115200,8,N,1,AVR 57600,8,N,1,AVR 38400,8,N,1,AVR	? ? ?	Status scanning SCAN_ERR_PARAMETERS SCAN_ERR_PARAMETERS SCAN_ERR_PARAMETERS	Interface sta S ITF_READ_S S ITF_READ_S S ITF_READ_S S ITF_READ_S	ETTINGS_CHECK_INTERF, ETTINGS_CHECK_INTERF, ETTINGS_CHECK_INTERF,	
	tatus Interface To Serial: COM4 To Serial: COM4	Parameters 115200,8,N,1,AVR 57600,8,N,1,AVR 18400,8,N,1,AVR 115200,8,N,1,AVR 57600,8,N,1,AVR 38400,8,N,1,AVR	? ? ? My Arduino My Arduino	Status scanning SCAN_ERR_PARAMETERS SCAN_ERR_PARAMETERS	Interface sta S ITF_READ_S S ITF_READ_S S ITF_READ_S ITF_READ_S JUND ITF_INTERFA ITF_INTERFA	ETTINGS_CHECK_INTERF, ETTINGS_CHECK_INTERF,	Update

Fig. 5.3. Serial communication setup: indicating successful check.

In the table above, you see in the column "Arduino board ID" the parameter that was part of the WawiLib.begin(...) statement in the Sketch (see Fig. 3.1.). Here you can keep track if you use multiple Arduino boards.

- ✓ Click right in the list view and select "Remove Inactive":
- ⇒ WawiLib will remove all interfaces that have failed their scan test.

Scan list + scan st	atus				
[V]=Active/Alias	Interface	Parameters	Board ID	Status scanning	Interface status
□ser1	🐌 Serial: COM4	115200,8,N,1,	AVR ?	SCAN_ERR_PARAMETERS	ITF_OFFLINE
□ser1	🐌 Serial: COM4	57600,8,N,1,A	AVR ?	SCAN_ERR_PARAMETERS	ITF_OFFLINE
□ser1	🛅 Serial: COM4	38400,8,N,1,A	AVR ?	SCAN_ERR_PARAMETERS	ITF_OFFLINE
□ser1	🖻 Serial: COM4	19200,8,N,1,A	VR ?	SCAN_ERR_PARAMETERS	ITF_OFFLINE
⊠ser2	🚺 Serial: COM26			SCAN_OK_ARDUINO_FOUND	ITF_OFFLINE
□ser2	🝈 Serial: COM26			SCAN_SKIPPED	ITF_OFFLINE
□ser2	🖻 Serial: COM26	-		SCAN_SKIPPED	ITF_OFFLINE
□ser2	🚺 Serial: COM26	1 Chang	e alias	SCAN_SKIPPED	ITF_OFFLINE
		Remov	/e		
<		Remov	e inactive		>

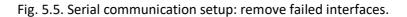
Fig. 5.4. Serial communication setup: remove failed interfaces.

- ✓ Press "OK" to close the "Scanning completed" dialog box.
- ✓ Press "OK" to close the "Automatic scan range settings" dialog box.

At this time, the connection parameters are identified.

- ✓ Activate "Display .Print() messages" and "Display communication protocol messages" window. (Click right and use the output window menu.)
- ✓ Press "Setup()" in the main window toolbar.

P New	Den E	₽ Save	😕 Print	Сору	∦ Cut	Paste	0 Offline	To Setup()	Loop()	₩rite all	► Continue) brkp	t brkpt				
	ilable variable	-			h	nterface/Ar	d. ID	Variable	e name	Actual va	lueWrite v	alue		Variab	le addre	ss and st	tatus
	er1\My Arduir			1		er1/My Ard	uino	delay	/On	500		6	delavOn=)x0202 [2]	ovtel x 1	VAR R	READI
	Single variab	le		2	5	er1/My Ard	uino	delay	Off	500			delayOff=				
	delayOff delayOff			3	1	er1/My Ard	uino	blinkCo	ounter	233		0	blinkCoun	ter=0x02C	3 [2 byte]	x 1 VA	AR_RI
	BlinkCoun	tor		4													
	Array	tter		5													
	, and y			6													
				7													
				<													>
ndex	Time			Node					Message								
917	10/07/2021	18:11:10	0.046	ser1/COM	26/My	Arduino			uC [02 58	50 4c 45 44	4 20 69 73	20 4f 4	e 2e 04 70] msg ok			
918	10/07/2021	18:11:10	0.0 <mark>4</mark> 6	ser1/COM	26/My	Arduino			LED is ON.								
919	10/07/2021	18:11:10	0.046	ser1/COM	26/My	Arduino			uC [02 59	50 0d 0a 0	4 0e] msg	ok					
920	10/07/2021	18:11:10	0.469	ser1/COM	26/My	Arduino			PC [01 bf 5	52 10 01 10	0 02 10 02	10 01 (00 10 02 10	01 c3 10 (02 04 2f <u>1</u>	l	
921	10/07/2021	18:11:10	0.500	ser1/COM	26/My	Arduino			uC [01	bf bf 06 f4	10 01 06 f	4 10 0 [.]	1 06 e9 00	04 50] OK	delay +-	31 ms	
922	10/07/2021	18:11:10	0.545	ser1/COM	26/My	Arduino			uC [02 5a	50 4c 45 44	4 20 69 73	20 4f 4	6 46 2e 04	3c] msg c	•k		
923	10/07/2021	18:11:10	0.545	ser1/COM	26/My	Arduino			LED is OFF								
924	10/07/2021	18:11:10	0.545	ser1/COM	26/My	Arduino			uC [02 5b	50 0d 0a 0	4 0c] msg	ok					
925	10/07/2021	18:11:10	0.591	ser1/COM	26/My	Arduino			PC [01 c0 !	53 04 93]							
926	10/07/2021	18:11:10	0.621	ser1/COM	26/My	Arduino			uC [01	c0 c0 06 10	01 10 02	40 00 4	40 00 ff 7f	ff 7f 10 04	10 02 05	00 10 01	1 00 -
927	10/07/2021	18:11:11	1.016	ser1/COM	26/My	Arduino			PC [01 c1 !	52 10 01 10	0 02 10 02	10 01 (00 10 02 10	01 c3 10	02 04 51]	
928	10/07/2021	18:11:11	1.061	ser1/COM	26/My	Arduino			uC [01	c1 c1 06 f4	10 01 06 f	4 10 0	1 06 e9 00	04 2e] OK	02		
929	10/07/2021	18:11:11	1.061	ser1/COM	26/My	Arduino			uC [02 de	lay +- 31 m	าร						
930	10/07/2021	18:11:11	1.061	ser1/COM	26/My	Arduino			WawiSrv.P	rint() demo	in loop() fu	unctior	, blinkcour	ter = 234			
931	10/07/2021	18:11:11	1.061	ser1/COM	26/My	Arduino			uC [02 5d	50 32 33 3	4 04 38] m	nsg ok					
932	10/07/2021	18:11:11	1.061	ser1/COM	26/My	Arduino					4 09] msg						
222	10/07/2021	18:11:11	1.062	ser1/COM	· · · ·				uC [02 5f 5	50 4c 45 44	20 69 73 2	20 4f 4	e 2e 04 77] msg ok			
933	10/07/2021	18:11:11	1.062	ser1/COM	26/My	Arduino			LED is ON.								. 1
933 934				ser1/COM							4 37] msg						



WawiLib will establish a connection to the Arduino board using the parameters as they were identified in the section above (connections to multiple boards at the same time are supported).

Notes:

- If you are using different types and families of Arduino boards, they all use different types of RTS/DTR handshake types. In order to resolve this, you can select different board families.
- Be careful (do not use) with 1200 baud as this can trigger a function on the Arduino Due that starts a firmware reset.
- When you open a "Serial Monitor Window" in the Arduino IDE, this can trigger a reset on some boards. A reset of your board can be triggered in the same way when you make a connection between WawiLib and your Arduino board.

• If you do not know which port to use for your Arduino connection, you can select all of them at the same time and press "add" and scan. The right combinations will light up in green after scanning.

6. Read and write variables with WawiLib

6.1. Watch variables

2	B		8	1	ж	(ît)	1	T	6	*	Þ	۲	×
New	Open	Save	Print	Copy	Cut	Paste	Offline	Setup()	Loop()	Write a	Continue	e brkpt	brkpt
	lable variable			Interfa	ce/Ard. ID	Variat	ole name	Act	ual value	w	rite <mark>valu</mark> e	Format	Variable address and status
	er1\My Ardui Single variat		1	ser1/M	y Arduino	blink	Counter		385			INT	@blinkCounter=0x01C7 [2 byte] x 1 VAR_READING_OK
	delayOn	Jie	2	ser1/M	y Arduino	de	layOn	0b0000'0	001 1111'(0100		BIT	@delayOn=0x0102 [2 byte] x 1 VAR_READING_OK -
	delayOff delayOff		3	ser1/M	y Arduino	del	ayOff	C	x01F4			HEX	@delayOff=0x0100 [2 byte] x 1 VAR_READING_OK -
	H blinkCour	nter	4	ser1/M	y Arduino	n	one					FLOAT	VAR_ERR_NOT_FOUND -
	Array		5										
ndex	Time			Node		Mes	sage						
55	25/07/2021	14:44:2	4.216	ser1/COM	18/My Ard.	. LED	is ON.						
56	25/07/2021	14:44:2	4.715	ser1/COM	18/My Ard.	. LED	is OFF.						
57	25/07/2021	14:44:2	5.226	ser1/COM	18/My Ard.	. Waw	/iSrv.Print() demo in l	loop() func	tion, blir	kcounter = 3	384	
58	25/07/2021	14:44:2	5.226	ser1/COM	18/My Ard.	. LED	is ON.						
59	25/07/2021	14:44:2	5.722	ser1/COM	18/My Ard.	. LED	is OFF.						
60	25/07/2021	14:44:2	6.237	ser1/COM	18/My Ard.	. Waw	/iSrv.Print(demo in l	loop() func	tion, blir	kcounter = 3	385	
61	25/07/2021	14:44:2	6.237	ser1/COM	18/My Ard.	. LED	is ON.						
62	25/07/2021	14:44:2	6.732	ser1/COM	18/My Ard.	. LED	is OFF.						

Fig. 6.1. Add variables to the grid using drag & drop.

- ✓ Go online (press Setup()) on the top toolbar.
- ✓ Drag the variables *blinkCounter* and *delayOn* from the tree control to the grid.
- ✓ Alternative: enter the names of the variables of interest in the grid.
- ✓ Modify the display format as indicated in Fig. 6.1.

The "Interface/Arduino ID" column will be filled in automatically as there is only 1 board active.

You can also click right on the grid and select "Available interfaces". Any active interfaces can be selected using this menu. This option is used to exchange data with multiple boards at the same time.

Do note the "Variable address and status" column: @delayOn=0x0102 means this variable is located at address 0x0102 in the Arduino board and its size is 2 bytes. The statement x 1 indicates that this variable is not an array. VAR_READING_OK is the status of the variable data exchange FSM.

Look at line 4: if a variable cannot be found in the Arduino board, the status of the FSM is VAR_ERR_NOT_FOUND.

6.2. Modify variables

In the upper window, you see the actual value of the variables. In the bottom window, you see the communication telegrams that are exchanged over USB with your Arduino board when it is online.

- ✓ In the output window, disable all output but enable "Display diagnostics messages".
- ✓ Fill in 100 as new value for *delayOff* in the write column.
- ✓ Press "Write all".
- ⇒ You see the actual value of delayOn change to 100 (in the upper window). The time the LED is on will change to 100ms.
- ⇒ In the bottom window, you see the result of your write action, if there is a format error, it will be displayed here.

2	B	8	1	x	🗈 🚺	1	6	₽	Þ	٠	×	
New	Open Save	Print	Сору	Cut	Paste Offline	e Setup()	Loop()	Write a	Continue	brkpt	brkpt	
	lable variables		Interfa	ce/Ard. ID	Variable name	Act	ual value	w	rite value	Format	Variable address	a
	r1\My Arduino Single variable	1	ser1/M	y Arduino	blinkCounter		637			INT	@blinkCounter=0x01C7 [2 byte] x 1 VAR_	R
T	delayOn	2	ser1/M	y Arduino	delayOn	0b0000'0	001 1111'(0100		BIT	@delayOn=0x0102 [2 byte] x 1 VAR_REA	DI
	delayOff delayOff	3	ser1/M	y Arduino	delayOff	0	x0100		100	HEX	@delayOff=0x0100 [2 byte] x 1 VAR_REA	D
	blinkCounter	4	ser1/M	y Arduino	none					FLOAT	VAR_ERR_NOT_FOUND -	
L	Array	5										>
Index	Time		Node		Message							
000	25/07/2021 14:47	:57.204	ser1/COM	18/My Ard.	LED is OFF.							
001	25/07/2021 14:47	:57.461	ser1/COM	18/My Ard	WawiSrv.Print	t() demo in l	oop() func	tion, blin	kcounter = 6	500		
002	25/07/2021 14:47	:57.461	ser1/COM	18/My Ard.	LED is ON.							
003	25/07/2021 14:47	:57.963	ser1/COM	18/My Ard	. LED is OFF.							
004	25/07/2021 14:47	:58.235	ser1/COM	18/My Ard	. WawiSrv.Print	t() demo in l	oop() func	tion, blin	kcounter = 6	501		
005	25/07/2021 14:47	:58.235	ser1/COM	18/My Ard.	LED is ON.							
006	25/07/2021 14:47	:58.446	ser1/COM	18/My Ard	. [Row 3] Writin	ng variable	delayOff=0	x0100 (H	IEX) complet	ted.		
007	25/07/2021 14:47	:58.732	ser1/COM	18/My Ard	. LED is OFF.							

Fig. 6.2. Change the value of the parameter delayOff.

In the bottom window you can find the following information:

- The target status of the communication interfaces is "Loop()".
- "Autowrite" is on.
- No data recorders were defined.
- No output recorders were defined.
- o ser1 corresponds to an Arduino named "My Arduino".
- COM 18 is the (virtual) USB serial port used.
- Baud rate = 115200 bits/second.
- \circ 8 data bits.
- N=No parity.
- 1=1 stop bit.
- AVR=AVR family type of board.
- The state of the communication interface Ser1 is ITF_LOOP.
- o 1539 message exchanges between PC and Arduino have been executed successfully.
- o 1539 message exchanges between PC and Arduino have been executed in total.

7. Record variable to file (introduction)

In this section, we will configure a data recorder to record the values of our parameters in an .xlsx file that is compatible with Microsoft Excel.

- ✓ Open the menu "Settings/Data Recording" in the main window.
- ✓ Press "Clear list".
- ✓ Select as data file format in the first tab: xlsx "Excel/LibreOffice compatible spreadsheet".
- ✓ Select "Overwrite current data file".
- ✓ Go to the second tab. (Fig.. 7.2.)
- ✓ Select 10 seconds as time base.
- ✓ Press "Add":

ata reco	order name:	REC1				
le prop	erties Record	details Disk	 usage and file size limitation 	tion		
Filenam	ne and directo	ry		Data file format		
Filenan	ne: WawiDat	taRecor <mark>ded</mark> .x	lsx	O csv: comma se	eparated values	
Directo	ory: C:\Users	\Johi\Docum	nents	🔿 xml: extensible	markup language	
		A		xlsx: Excel/Libre	eOffice compatible	spreadshee
	going online o			CSV separator (\t=1	tab)	
• Ove	erwrite <mark>curr</mark> ent	data file		CSV Separator (\t=)	(dD)	
\sim	18 20100	27.02				
_			current data file			
			current data file tart date and time to filena	ame)		
) Star		ta file (add st	tart date and time to filena	ame) Dir	Add record tyr	Add
) Star	t with new da red data recor File mode	ta file (add st ders Time base	tart date and time to filena		Contraction and a second second second	
) Star	t with new da red data recor File mode	ta file (add st ders Time base	tart date and time to filena	Dir	Contraction and a second second second	Add Remove Update

Fig. 7.1. Define a new data recorder.

Data recording settings				×
Data recorder name: REC1				
File properties Record details Dis	k usage and file size limitat	tion		
Recording triggers				
✓ Record time based (for chang	e based: modify propertie	s of variable in main table)		
Recording interval (sec): 10				
Data record type				
Add record type (time based	(change based)			
Data record timestamp settings				
Add date	Add date (UTC)			
✓ Add time	Add time (UTC)			
Add milliseconds (approx.)	Add relative timestam	np (elapsed time)		
Configured data recorders				
Name File mode Time base		Dir	Add record typ	Add
REC1 OVERWRITE 10 sec	WawiDataRecorded.xlsx	C:\Users\Johi\Documents	no	Remove
				Update
				Clear list
<			>	
	OK Car	Default para	meters	

Fig. 7.2. Define a new data recorder time base = 10 sec.

This will create a data recorder in line with your actual settings.

- ✓ Press "OK" to close the dialog box.
- ✓ In the main grid, select all variables recorder fields (Fig. 7.3 in blue), click right and select "Available data recorders/Rec1":

P New	Den 🖻	₽ Save	🖶 Print	Сору	<mark>Ж</mark> Cut	🗈 Paste	Offline	To Setup()	Loop()	↓ Write all ⊂	• ontinue	e brkpt	⊠ brkpt			
	able variable		^		Interfa	ce/Ard. ID	Va	riable na	me	Actual value	Recorde	r	Var	iable addres	s an <mark>d</mark> status	
	r1\My Ardui			1	ser1/M	ly Arduino		delayOn		500		@del	ayOn=0x0202	[2 byte] x 1	- VAR READING	i Oł
1	Single varial	bie		2	ser1/M	ly Arduino		delayOff		500		@del	avOff=0x0200	[2 bvte] x 1 -	- VAR READING	5 01
	delayOff delayOff delayOff			3	ser1/M	ly Arduino	b	linkCount	er	376	1				1 VAR READ	-
	BlinkCour	nter		4									terfaces	>		
1	Arrav	ice.	¥ 4								Ava	ailable d	ata recorders	>	REC1	
ndex	Time			Node		Messa	ge				Dis	play for	mats	>		
1015	11/07/2021	9:24:57.	830	ser1/CON	126/My Ar	d WawiS	rv.Print()	demo in l	oop() fur	ction, blinkcou	n Off	line				
1016	11/07/2021	9:24:57.	830	ser1/CON	126/My Ar	d LED is	ON.				Set	up()				
1017	11/07/2021	9:24:58.	342	ser1/CON	126/My Ar	d LED is	OFF.				✓ Loc	1.2				
1018	11/07/2021	9:24:58.	840	ser1/CON	126/My Ar	d WawiS	irv.Print()	demo in l	oop() fur	ction, blinkcou	n	ite row		Ctrl+W		
1019	11/07/2021	9:24:58.	840	ser1/CON	126/My Ar	d LED is	ON.						ed fields	Ctri+vv		
1020	11/07/2021	9:24:59.	339	ser1/CON	126/My Ar	d LED is	OFF.						ed fields			
	11/07/2021				126/My Ar			demo in l	oop() fur	ction, blinkcou	wri	ite all		Alt+W	-	
	11/07/2021				126/My Ar		1. TO 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.				Cut			Ctrl+X		
	11/07/2021				126/My Ar						Cor	р <mark>у</mark>		Ctrl+C		
1024	11/07/2021				126/My Ar			demo in l	oop() fur	ction, blinkcou	n Pas	te		Ctrl+V		- 1
1025	11/07/2021	9:25:00.	860	ser1/CON	126/My Ar	d LED is	ON.				Cle	ar row(s)			
<											Inse	ert row				>
				Loop() Au	itowrite or	REC1 [F	RECO_NO	_TAGS] cn	t=0 s	er1=My Arduin	o Ins	ert rows			ok/tot: 3736/37	736
											Del	ete row((s)			
											Cle	ar entire	table			
											Var	iable pro	operties			
											Res	et View				

Fig. 7.3. Link all the variables to a data recorder REC1.

- ✓ Disable the "Trace protocol" option in the output window.
- ✓ Enable "Automatic scroll" & "Display data recording"
- ✓ Disable all other options ad indicated in Fig. 7.4.:

연 New	Den 🕞	₽ Save	👼 Print	Copy	ょ Cut	Daste Paste	0ffline	Setup()	Toop()	Write all	▶ Continue	• brkpt	⊠ brkpt	
1	lable variable		^		Interf	ace/Ard. ID	V	ariable na	me	Actual value	Record	er	Variable address and status	
1	r1\My Ardui Single varial			1	ser1/I	My Arduino		delayOn		500	REC1	@de	elayOn=0x0202 [2 byte] x 1 VAR_READING	0
	delavOn	JIC .		2	ser1/I	My Arduino		delayOff		500	REC1	@de	elayOff=0x0200 [2 byte] x 1 VAR_READING	_0
1	delayOff delayOff			3	ser1/I	My Arduino	I	olinkCount	er	488	REC1	@bli	linkCounter=0x02C3 [2 byte] x 1 VAR_REA[NIC
	B blinkCour	nter		4										
	Arrav		× <	۲										>
ndex	Time			Node			Message	(
000	11/07/2021			REC1	Dis	play .print()	message	s						
201	11/07/2021			REC1 REC1	Dis	play diagno	stics mes	sages						
002 003	11/07/2021			REC1		splay commu		-	noscado					
003	11/07/2021			REC1				protocori	nessaye	,				
004	11,01,2021	5.20.55.	000	NECT	V Dis	splay data re	cording							
					Dis	splay output	window	recording						
					Co	py selected	text			Ctrl+C				
					Cle	ar Window								
					🗸 Au	tomatic scro	11							
					Re	set view								

Fig. 7.4. Enable display the recorded data in the output window.

✓ Press "Setup()".

You will now see the different values of your variables as they are written to the .xlsx file in the output window.

✓ Press "Offline".

2	B		8	0	X	6	1	6	6	÷		• •
New	Open	Save	Print	Сору	Cut	Paste	Offline	Setup()	Loop()	Write all C	Continue b	rkpt brkpt
	able variable				Interf	ace/Ard. ID	V	ariable na	me	Actual value	Recorder	Variable address and status
	1\My Ardui Single varia			1	ser1/M	Arduino		delayOn		500	REC1	@delayOn=0x0202 [2 byte] x 1 VAR_OFFLINE - REC_A
	🗄 delayOn	JIE		2	ser1/M	Arduino		delayOff		500	REC1	@delayOff=0x0200 [2 byte] x 1 VAR_OFFLINE - REC_F
	■ delayOff			3	ser1/M	/y Arduino	ł	olinkCount	er	880	REC1	@blinkCounter=0x02C3 [2 byte] x 1 VAR_OFFLINE - R
	blinkCour	oter		4								
	Array			5								
				6								
				7								
				8								
				9								
				10	_							
				11	_							
			<									>
ndex	Time			Node		Messa	ge					
000	11/07/202	9:33:23.51	8	REC1		openii	ng file C:\	Users\Joh	i∖Docum	nents\WawiData	Recorded.cs	sv for overwrite OK
001	11/07/202	9:33:23.51	8	REC1		write h	neader: d	ate time d	elayOn (INT) delayOff (II	NT) blinkCou	unter (INT)
002	11/07/202	9:33:24.06	3	REC1		rec(o.t	.): 11/07,	2021 9:33	:24 500	500 877		
003	11/07/202	9:33:25.02	9	REC1		rec(o.t	.): 11/07,	/2021 9:33	:25 500	500 878		
004	11/07/202	9:33:26.06	1	REC1		rec(o.t	.): 11/07,	/2021 9:33	:26 500	500 879		
005	11/07/202	9:33:27.03	5	REC1		rec(o.t	.): 11/07,	2021 9:33	:27 500	500 880		
006		9:33:27.63		REC1						file closed: offli		
007	11/07/202	9:33:27.63	2	RFC1		closing	n openeo	file C:\Us	ers\Johi\	Documents\Wa	awiDataReco	orded.csv OK

Fig. 7.5. After going offline, the output window displays the name of the file created.

0	Hor	me Inse	ert Page Layout	Formulas Da	ata	Review	View	Developer	@ .	- 🗇 🗄
Par	ste	Calibri BI		= = = · ·	Genera	%,	A Styles	Insert *	1100	Find & Select
lip	boa 🖻	Fo	ont 🕞 .	Alignment 🔤	Numb	ber 「a		Cells	Editi	ng
	E1:	1	• (*	f_{x}						
	1	4	В	C			D	E		F
1	date		time	delayOn (INT)		delayC	off <mark>(INT</mark>) blinkCour	nter (INT)	-
2	11/0	7/2021	9:33:24		500		50	D	877	
3	11/0	7/2021	9:33:25		500		50	D	878	
4	11/0	7/2021	9:33:26		500		50	D	879	
5	11/0	7/2021	9:33:27		500		50	D	880	
6	11/0	7/2021	9:33:28	file closed: off	line					
-		1	ataRecorded	A-1	_			100%		

✓ Open the recorded .xlsx file in Microsoft Excel or equivalent.

Fig. 7.6. Data file opened in Microsoft Excel.

	WawiDataReco	rded.csv -	LibreOffice Calc		-		٦
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>I</u> nsert	F <u>o</u> rmat St <u>y</u> les <u>S</u> r	neet <u>D</u> ata <u>T</u> oo	ls <u>W</u> indow <u>H</u> elp		×
	• 📄 • 🔒	- 🔒 🕯	a 🛛 🕹 🔓	• 🖌 🤌 😽		-	»
Libe	eration Sans	× 10	→ B I <u>U</u>	A • 🔊 • 🖹			>>
A1		~ f x Σ	• = date			▼ :	:
	A	В	С	D	E	F 🔨	
1	date	time	delayOn (INT)	delayOff (INT)	blinkCounter (INT)		ŧ
2	11/07/2021	9:33:24	500	500	87	7	
3	11/07/2021	9:33:25	500	500	87	8 8	<u>4</u>
4	11/07/2021	9:33:26	500	500	87	9 📻	a
5	11/07/2021	9:33:27	500	500	88		4
<						> (4	
K.	< > > +	Wawi[DataRecorded			Ŧ	
Shee	et 1 of 1 D	efault	English (USA)	🛱 🛛 Average	e: ; Sum: 0 🕴 — ——	+ 1009	%

Fig. 7.7. Data file opened in LibreOffice Calc.

You can see the date, the time, the relative timestamp, the type of record (on timer or on change recording), the name of the variables and their value in the Excel table.

8. Record .print() output to file (introduction)

In this section, we will configure an output recorder to record the output of WawiSrv.print() statements to a .txt file.

- ✓ Open the menu "Settings/Output Recording" in the main window.
- ✓ Press "Clear list".
- ✓ Select as data file format in the first tab: "csv: comma separated values".
- ✓ Select "Overwrite current data file".
- ✓ Go to the second tab.
- ✓ Select Arduino WawiSrv.print() messages (see Fig. 8.2.)
- ✓ Press "Add":

Output recording settings				×
Output recorder name: OREC1				
File properties Output message recording details Disk usage a	and file size lim	itation Arduino bo	oards/interfaces to record	
File name and directory	Data	file format		
Filename: WawiOutputRecorded.csv	• c	sv: comma separat	ed values	
Directory: C:\Users\Johi\Documents	() x	ml: extensible mark	kup language	
When going online on Arduino:	Ox	sx: Excel/LibreOffic	e compatible spreadsheet	
• Overwrite current data file	CSV s	eparator (\t=tab)	•	
\bigcirc Append new data records to current data file				
○ Start with new data file (add start date and time to filenam	lej			
Configured output recorders				Add
Name File mode File Dir OREC1 OVERWRITE WawiOutputRecorded.csv C:\Users\Jo	hi\Documents	21	Add Ard. WawiSrv.printf() yes	Remove
	in (Bocaments	110	yes	Update
				Clear list
<			>	
ОК	Cancel	Default pa	rameters	

Fig. 8.1. Define a new output recorder.

Output recording settings	×
Output recorder name: OREC1	
File properties Output message recording details Disk usage and file size limitation Arduino boards/interfaces to record	
Date record timestamp settings	
Add date Add time (UTC)	
Add time Add relative timestamp (elapsed time)	
Add milliseconds (approx.)	
Add source and record type to file	
Add record type (Node name/PRINT, /PROT, /DIAG, /ERR, /DATA)	
Record output message types (Arduino board & interface related) Record output message types (general messages) Arduino "WawiSrv.print()" messages WawiLib disagnostics messages Disagnostics messages WawiLib data recording messages Communication protocol messages WawiLib error messages Error messages WawiLib error messages	
Configured output recorders	Add
Name File mode File Dir Add record type Add Ard. WawiSrv.printf()	Remove
OREC1 OVERWRITE WawiOutputRecorded.csv C:\Users\Johi\Documents no yes	Update
	Clear list
< >>	
OK Cancel Default parameters	

Fig. 8.2. Select subset of output recorder messages to be recorded.

- ✓ Press "OK" to close the dialog box.
- ✓ Enable "Display output window recording () and "Display .print() messages".

New	🖻 Open	■ Save	😕 Print	Copy	∦ Cut	🗈 Paste	The offline	Setup()	Loop()		▶ Continue	e brkpt	⊠ brkpt			
Availa	able variable	s	<u>^</u>	copy		ace/Ard. ID		ariable na		Actual value			bitpt	Variable a	ddress aı	nd stat
1.	1\My Ardui			1	ser1/N	Arduino		delavOn		500		@de	avOn=0x	0202 [2 byte	x 1 VA	R REA
	Single variat ∃delayOn	ole		2	ser1/N	My Arduino		delayOff		500				0200 [2 byte		
1.1	■ delayOn ■ delayOff			3	ser1/N	My Arduino	1	blinkCount	er	2878			12	r=0x02C3 [2		Concernance of
	B blinkCoun	tor	× <													>
dex 1	Time			Node			Messag	le								-
55 1	11/07/2021	10:06:50	.265	ser1/COM	26/My A	rduino	LED is (DN.								
56 1	11/0//2021	10:06:50	.325	OREC1			rec. out	put: 11/0/	/2021 10):06:50 WawiS	✓ Displa	ay .print) messag	es		
57 1	11/07/2021	10:06:50	.325	OREC1			rec. out	put: 11/07	/2021 10):06:50 LED is	 Displa 	ay diagn	ostics me	ssages		
58 1	11/07/2021	10:06:50	.761	ser1/COM	26/My A	rduino	LED is (OFF.			Displa	ay comr	nunication	n protocol m	essages	
59 1	11/07/2021	10:06:50	.805	OREC1			rec. out	put: 11/07	/2021 10):06:51 LED is	Dical	w data i	ecording	· · · · · · · · · · · · · · · · · · ·		
70 1	11/07/2021	10:06:51	.274	ser1/COM	26/My A	rduino	WawiSr	v.Print() de	mo in lo	op() function,		*				
71 1	11/07/2021	10:06:51	.274	ser1/COM	26/My A	rduino	LED is (DN.			✓ Displa	ay outpu	it window	recording		
72 1	11/07/2021	10:06:51	.289	OREC1			rec. out	put: 11/07	/2021 10):06:51 WawiS	Сору	selected	text			Ctrl+
73 1	11/07/2021	10:06:51	.289	OREC1			rec. out	put: 11/07	/2021 10):06:51 LED is	Clear	Window	1			
74 1	11/07/2021	10:06:51	.772	ser1/COM	26/My A	rduino	LED is (OFF.			 Autor 	matic sc	oll			
75 1	11/07/2021	10:06:51	.831	OREC1			rec. out	put: 11/07	/2021 10	0:06:52 LED is	Reset	ulaur				

Fig. 8.3. Enable Display print() messages & Display output window recording.

✓ Press "Setup()".

You will now see the output of the print() statements in the output window. The output is generated in your sketch by the lines indicated in yellow in Fig. 24.

Do note (Fig.. 8.5.) that the output itself is displayed and what is written to file as well. (See Node column.) The settings of the output window do not influence the recoding itself.

✓ The LED on your board should blink 500ms on and 500ms off.

```
#include<WawiSerialUsb.h>
WawiSerialUsb WawiSrv;
#defineLED 13
// test variables for demo:
int delayOn = 500;
int delayOff = 500;
int blinkCounter = 0;
// make variables of interest knownto WawiLib:
// thisfunction is used in WawiSrv.begin(....)
void wawiVarDef()
{
    WawiSrv.wawiVar(delayOn);
    WawiSrv.wawiVar(delayOff);
    WawiSrv.wawiVar(blinkCounter);
void setup()
{
    Serial.begin(115200);
    WawiSrv.begin(wawiVarDef, Serial, "My Arduino");
    pinMode(LED, OUTPUT);
}
void loop()
{
    blinkCounter++;
    WawiSrv.print("WawiSrv.Print() demo in loop() function, blinkcounter = ");
   WawiSrv.println(blinkCounter);
    WawiSrv.println("LED is ON.");
    digitalWrite(LED, HIGH);
    WawiSrv.delay(delayOn);
    WawiSrv.println("LED is OFF.");
    digitalWrite(LED, LOW);
    WawiSrv.delay(delayOff);
    WawiSrv.loop();
```

Fig. 8.4. Minimal Arduino example WawiBlinkSerial.ino

✓ Press "Offline".

ମ୍ଭ New	Dpen	F Save	Print	Сору	ん Cut	🖻 Paste	Offline	To Setup()	Toop()	↓ Write all ⊂		• 🐞 rkpt brkpt	
	lable variable		^		Interfa	ace/Ard. ID	Va	ariable na	me	Actual value	Recorder		Variable a
2	er1\My Arduii			1	ser1/N	/ly Arduino		delayOn		500		@delayOn=0x02	202 [2 bvte
	Single variab	bie		2	1000	Arduino		delayOff		500		@delayOff=0x0	and the second second
	delayOff	tor	~ <	3		Ay Arduino	ł	olinkCount	er	3641		@blinkCounter=	
ndex	Time			Node	Me	ssage							
3701	11/07/2021	10:19:32	2.441	OREC1	rec	output: 11/	07/2021	10:19:32 L	ED is O	N.			
3702	11/07/2021	10:19:32	2.921	OREC1	rec	output: 11/	07/2021	10:19:33 L	ED is O	FF.			
3703	11/07/2021	10:19:33	8.467	OREC1	rec	output: 11/	07/2021	10:19:33 \	VawiSrv	.Print() demo in	loop() funct	tion, blinkcounter	= 3635
3704	11/07/2021	10:19:33	8.467	OREC1	rec	output: 11/	07/2021	10:19:33 L	ED is O	N.			
3705	11/07/2021	10:19:33	8.949	OREC1	rec	output: 11/	07/2021	10:19:34 L	ED is O	FF.			
370 <mark>6</mark>	11/07/2021	10:19:34	1.432	OREC1	rec	output: 11/	07/2021	10:19:34 \	VawiSrv	.Print() demo in	loop() funct	tion, blinkcounter	= 3636
3707	11/07/2021	10:19:34	.432	OREC1	rec	output: 11/	07/2021	10:19:34 L	ED is O	N.			
3708	11/07/2021	10:19:34	.975	OREC1	rec	output: 11/	07/2021	10:19:35 L	ED is O	FF.			
3709	11/0//2021	10:19:35	.460	OREC1	rec	output: 11/	07/2021	10:19:35 \	VawiSrv	.Print() demo in	loop() funct	tion, blinkcounter	= 3637
3710	11/07/2021	10:19:35	5.460	OREC1	rec	output: 11/	07/2021	10:19:35 L	ED is O	N.			
3711	11/07/2021	10:19:35	5.945	OREC1	rec	output: 11/	07/2021	10:19:36 L	ED is O	FF.			
3712	11/07/2021	10:19:36	5.486	OREC1	rec	output: 11/	07/2021	10:19:36 \	VawiSrv	.Print() demo in	loop() funct	tion, blinkcounter	= 3638
3713	11/07/2021	10:19:36	5.486	OREC1	rec	output: 11/	07/2021	10:19:36 L	ED is O	N.			
3714	11/07/2021	10:19:36	5.970	OREC1	rec	output: 11/	07/2021	10:19:37 L	ED is O	FF.			
3715	11/07/2021	10:19:37	.454	OREC1	rec	output: 11/	07/2021	10:19:37 \	VawiSrv	.Print() demo in	loop() funct	tion, blinkcounter	= 3639
3716	11/07/2021			OREC1	rec	output: 11/	07/2021	10:19:37 L	ED is O	N.			
3717	11/07/2021	10:19:37	.997	OREC1		output: 11/							
3718	11/07/2021	10:19:38	8.483	OREC1	rec	output: 11/	07/2021	10:19:38 \	VawiSrv	.Print() demo in	loop() funct	tion, blinkcounter	= 3640
3719	11/07/2021			OREC1	rec	output: 11/	07/2021	10:19:38 L	ED is O	N.			
3720	11/07/2021			OREC1		output: 11/							
3721	11/07/2021			OREC1		1 - C					loop() funct	tion, blinkcounter	= 3641
3722	11/07/2021			OREC1		output: 11/							
3723	11/07/2021			OREC1		output: 11/	07/2021	10:19:40 L	ED is O	FF.			
3724	11/07/2021			OREC1		te footer:							
3725	11/07/2021	10:19:40).475	OREC1	clo	sing opened	file C:\U	sers\Johi\[ocume	nts\WawiOutpu	tRecorded.cs	sv OK	

Fig. 8.5. Arduino output recording displayed in window.

✓ Open the file WawiOutputRecordd.csv from your "Documents" folder.

	B1	- (*	<i>f</i> ₅ time	2					*
	А	В	С	D	E	F	G	Н	
24	11/07/2021	10:19:37	WawiSrv.P	rint() dem	o in loop()	function,	blinkcount	er = 3639	
25	11/07/2021	10:19:37	LED is ON.						
26	11/07/2021	10:19:38	LED is OFF						
27	11/07/2021	10: <mark>19:3</mark> 8	WawiSrv.P	rint() dem	o in loop()	function,	blinkcount	er = 3640	
28	11/07/2021	10: <mark>1</mark> 9:38	LED is ON.						
29	11/07/2021	10:19:39	LED is OFF						
30	11/07/2021	10:19:39	WawiSrv.P	rint() dem	io in loop()	function,	blinkcount	er = 3641	
31	11/07/2021	10:19:39	LED is ON.						
32	11/07/2021	10: <mark>19:</mark> 40	LED is OFF						
33	11/07/2021	10: <mark>1</mark> 9:40	End record	ling.					
34									-
◀	▶ ▶ WawiOut	putRecord	led 🖉 🦳			Ш		•	
Ready						100%	9 () .:

Fig. 8.6. Arduino output recording file opened in Excel.

If you like, you can write also to an XML database file and WawiLib also supports closing the file each hour so your recordings remain limited in file size.

9. Introduction to WawiLib breakpoints

Sometimes you want your code to stop at a certain point. Advanced debuggers have these functions standard. WawiLib is no substitute for these tools. However sometimes a simple breakpoint can come in handy. Therefor WawiLib contains a basic breakpoint functionality.

- ✓ Open the example File\Examples\WawiSerialUsb\WawiSerialUsbBreakpoint.ino in the IDE.
- ✓ Compile and download the example.
- ✓ Connect WawiLib the board using "Settings\Communication interfaces" as in §5.
- ✓ Press "Setup()".

```
#include <WawiSerialUsb.h>
WawiSerialUsb WawiSrv;
#define LED 13
// test variables for demo:
int delayOn = 500;
int delayOff = 500;
int blinkCounter = 0;
bool led;
// make variables of interest known to WawiLib:
// this function is used in WawiSrv.begin(....)
void wawiVarDef()
{
    WawiSrv.wawiVar(delayOn);
    WawiSrv.wawiVar(delayOff);
    WawiSrv.wawiVar(blinkCounter);
    WawiSrv.wawiVar(led);
}
void setup()
{
    Serial.begin(115200);
    WawiSrv.begin(wawiVarDef, Serial, "My Arduino");
    pinMode(LED, OUTPUT);
    WawiSrv.wawiBreakDisable();
}
void loop()
{
    blinkCounter++;
    WawiSrv.print("WawiSrv.Print() demo in loop() function, blinkcounter = ");
    WawiSrv.println(blinkCounter);
    WawiSrv.println("LED is ON.");
    led = HIGH;
    digitalWrite(LED, led);
    WawiSrv.delay(delayOn);
    if (blinkCounter % 5 == 0)
        WawiSrv.wawiBreak(1, "Break after led is on");
    WawiSrv.println("LED is OFF.");
    led = LOW;
    digitalWrite(LED, LOW);
    WawiSrv.delay(delayOff);
    if (blinkCounter % 10 == 0)
        WawiSrv.wawiBreak(2, "Break after led is off");
    WawiSrv.loop();
```

Fig. 9.1. WawiLib breakpoint support demo.

✓ Add the variables to the grid as indicated in Fig. 9.2.

New	Dpen Save	Print	t Copy	Ж Cut	Deste Paste	Offline	Setup()	Loop()	Write all	► Continue	brkpt	🚫 brkpt				
-	able variables			Inter	face/Ard. I	D	V	ariable nar	ne	Actual	value	Write value	Format	1		
ia ser1\My Arduino ia Single variable			1/		My Arduino			blinkCount	er	39			INT			
e- delayOn		ngle variable		D		delayOff		500	0		INT					
	⊕ delayOff	yon				INT										
	H- blinkCounter		4	ser1/	My Arduine	D		delayOn		500			INT			
	+- led		5													
	Array		6													
		<												>		
Index	Time		Node		Message									1		
014	25/07/2021 11:42:5	3,349	ser1/COM18/N	Arduino	LED is OFF.											
015	25/07/2021 11:42:5		ser1/COM18/N	-												
016		25/07/2021 11:42:58.863 ser1/COM18/My Arduino														
017					LED is OFF.											
018	25/07/2021 11:42:5	5/07/2021 11:42:59.873 ser1/COM18/My Arduino			WawiSrv.	WawiSrv.Print() demo in loop() function, blinkcounter = 28										
019	25/07/2021 11:42:59.873 ser1/COM18/My Ard		Arduino	LED is ON.												
020	25/07/2021 11:43:00.371 ser1/COM18/My Arc		Arduino	LED is OFF.												
021	25/07/2021 11:43:00.884 ser1/COM18/My Arduin			Arduino	no WawiSrv.Print() demo in loop() function, blinkcounter = 29									- 1		
022	25/07/2021 11:43:00.884 ser1/COM18/My Arduino			LED is ON.												
023	25/07/2021 11:43:0	1.382	ser1/COM18/M	Arduino	LED is OF	F.										
	25/07/2021 11:43:0	1.898	ser1/COM18/N	Arduino	WawiSrv.Print() demo in loop() function, blinkcounter = 30											
024	25/07/2021 11:43:0	1.898	ser1/COM18/N	Arduino	LED is ON	۷.										
	25/07/2021 11:43:00	2.411	ser1/COM18/N	Arduino	Breakpoi	nt 1 hit: Bre	ak after led	is on								
025	23/01/2021 11.43.00		1/COM410/A	Arduino	file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkUsbBreakpoint\WawiBlinkUsbBreakpoint.ino											
025 026	25/07/2021 11:43:0	2.411	seri/COMTO/M													
024 025 026 027 028			ser1/COM18/N		function:	loop, line:	62									

Fig 9.2. Variables of WawiBlinkUsbBreakpoint added to grid.

- ✓ Press "brkpt" in the toolbar.
- ⇒ The sketch will run further until blinkCounter is a multiple of 5 or 10 and then show a message in the output window as indicated in figure 29.
- ⇒ The output window contains the line and a message you defined yourselves in your code.

	•	• •	
⇒	<pre>if (blinkCounter % 5 == 0)</pre>		
⇒	WawiSny wawiBreak(1 "Break	after led is on"	<u>۱</u> .

- ⇒ The output window also contains the source file, the function and the source line where the breakpoint was hit.
- ✓ Press "continue" in the toolbar.
- ✓ The sketch will run further another breakpoint is hit.

New	Dpen 🔁	F Save	Print	Сору	∦ Cut	Paste	Offline	Setup()	Loop()	₩rite all	Continue	b rkpt	⊠ brkpt		
Available variables in ser1\My Arduino					Interface/Ard. ID			Variable name			Actual value		Write value	Format	
				1 ser1/My Arduit			rduino blinkCounter				17	5		INT	7
Single variable						50	D		INT						
	delayOff delayOff			3	ser1/	My Arduino			led		1			INT	
	+- blinkCounter			4	ser1/	My Arduino			delayOn		50	D		INT	
	+ led			5											
	Array			6											
			<	1.											>
Index	Time		N	ode		Message									
000	25/07/2021 11:47:34.820		820 se	er1/COM18/	My Arduino	WawiSrv.Print() demo in loop() function, blinkcounter = 175									
001	25/07/2021 11:47:34.820 ser1/COM18		820 se	er1/COM18/	My Arduino	Ay Arduino LED is ON.									
002			349 se	er1/COM18/	My Arduino	y Arduino Breakpoint 1 hit: Break after led is on									
003			349 se	er1/COM18/	My Arduino	duino file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkUsbBreakpoint\WawiBlinkUsbBreakpoint							bBreakpoint.i	no	

Fig 9.3. WawiBlinkUsbBreakpoint hit a breakpoint again.

10. Further reading

This demo demonstrates the concept of WawiLib using the USB programming port of your Arduino board. WawiLib has more extended functions that are presented in other demos. Functions of interest to you can be the monitoring and modification of strings or the use of various representation formats (HEX/INT/UINT/CHAR/STRING/FLOAT/DOUBLE).

Arrays of variables are also supported with WawiLib. Recording of variables can be executed "on change", "on timer" or both. Data recording can also be done with one file per hour or per day to make the generated files more manageable.

In the same way WawiLib supports recording of the output of .print() statements to a file on the disk of the PC. Files remain manageable as they can also be saved per hour or per day.

WawiLib also supports an elementary breakpoint facility that can be very handy debugging smaller Arduino's that have no on-board debug support or by absence of a special cable.

WawiLib supports links via Wi-Fi, cabled Ethernet, hardware serial, software serial and via USB to serial converters.

Arrays of variables are also supported by WawiLib. Recording variables can be executed "on change", "on timer" or both. Data recording can also be done with one file per hour or per day to make the generated files more manageable. WawiLib supports links via USB, Wi-Fi, cabled Ethernet, RS232C, hardware serial, software serial and via USB to serial converters.

I hope you enjoyed this demo. Visit us on <u>www.sylvestersolutions.com</u> for more demos.