Recording variables and .print() output with WawiLib

Contents

1.	Intro	oduction	2
1.:	1.	Objective of this document	2
1.2	2.	Software and hardware requirements	2
1.3	3.	Required user experience	2
2.	The	"WawiRecUSB" Demo sketch example	3
2.3	1.	Concept of "WawiBlinkRecUSB"	3
2.2	2.	Download and execute "WawiBlinkRecUSB"	3
3.	Wav	wiLib data recording to disk file	6
3.:	1.	Introduction	6
3.2	2.	Data recorders	6
3.3	3.	Data recording trigger aspects	9
3.4	4.	Data recording detailed example	10
3.0	6.1.	Recorded variables	10
3.0	6.2.	Time based recording of a variable or a series of variables	11
3.0	6.3.	Change based recording of a variable or a series of variables	15
3.5	5.	Data recorder storage aspects	18
4.	Wav	wiLib .print() recording to disk file	21
4.:	1.	Introduction	21
4.2	2.	Define an output recorder	21
5.	Fina	I notes	27
6.	Furt	her reading	27



1. Introduction

1.1. Objective of this document.

The first objective of this document is to describe how to use WawiLib to record variables. Points of interest are triggers to write data records to a disk and how to limit the sizes of the recorded datafiles. WawiRecUsb.ino, a demo sketch supplied with the WawiSerialUsb library, will be used to explain the concepts.

The second objective of this document is to describe how to use WawiLib to record .print() debug output to a PC disk file. WawiRecOutput.ino, a demo sketch supplied with the WawiSerialUsb library, will be used to explain the concepts.

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.

You also need a program to open the recorded data files: Excel or OpenOffice to open .xlsx files, a text editor to open .csv files and an XML viewer (I use XML Notepad from Microsoft) to open .xml files.

In this demo, the USB programming port of the Arduino is used as the communication interface between WawiLib and your Arduino shield. This demo can easily be converted to other types of communication links. The WawiLib getting started demos for serial, Ethernet and Wi-Fi communication can be used as a base for the conversion to another interface type. The only thing you have to do is replace the initialization code as in the demo's for WawiWifi and WawiEthernet.

The hardware you need is an Arduino board, a USB programming cable, 3 Dupont male-male (breadboard) wires and a Windows PC (32 or 64 bit). In this demo, we will use the Arduino UNO board but other boards can be used in a similar or even identical way.

The data recorder that is part of WawiLib is typically used to record all kinds of I/O signals. In this demo, we will record mainly internal variables of the Arduino. This choice was made to keep things simple and easy to understand. On <u>www.sylvestersolutions.com</u>, you will find application notes that enable you to build other applications using specific hardware.

1.3. Required user experience

You should have completed the tutorial "Getting started with WawiLib USB" and "Debugging with WawiLib USB". There are no specific additional requirements.

2. The "WawiRecUSB" Demo sketch example

2.1. Concept of "WawiBlinkRecUSB"

This application builds further on the sketch "WawiBlinkDebugUsb" from the demo "Debugging with WawiLib". WawiBlinkRecUsb is in fact WawiBlinkDebugUsb with its errors corrected.

The objective of the program is to blink 5 sec after a pulse on digital input 5, to blink 7 sec after a pulse on input 6 and to blink 10 seconds after a pulse on input 7. If one of the IO's remains high, blinking continues.

The program prints diagnostic output to the WawiLib output window when the LED is blinking:

- Counting down:
- The value of *blinkTimeActual*;
- LED is ON.
- LED is OFF"

2.2. Download and execute "WawiBlinkRecUSB"

- ✓ Open the demo sketch using the menu "File\Examples\WawiSerialUsb\WawiBlinkRecUSB" in the Arduino IDE.
- ✓ Connect inputs 5, 6 and 7 to the GND pins of your board.
- ✓ Compile and download WawiBlinkRecUSB to your Arduino board.

```
/*
* Project Name: WawiRecUsb
* File: WawiRecUsb.ino
* Detailed manual:
* www.SylvesterSolutions.com\documentation -> "Recording variables with WawiLib.pdf"
* Description: demo file library for WawiSerialUsb library.
* Data recorder demo.
* => Record values of variables to disk
* => Record .print() output to disk
\ast Use the USB programming port to make connection with the Arduino board.
* Variables can be checked & modified with the WawiLib-PC software.
* Author: John Gijs.
* Created March 2020
* More info: www.sylvestersolutions.com
* Technical support: support@sylvestersolutions.com
* Additional info: info@sylvestersolutions.com
*/
#include <WawiSerialUsb.h>
WawiSerialUsb WawiSrv;
#define LED 13 // blinking light
#define IN_5 5 // light start blinking switch 1
#define IN_6 6 // light start blinking switch 2
#define IN_7 7 // light start blinking switch 3
// variables for demo:
long int blinkTimeActual = 0; // counter blink active (milliseconds)
long int blinkTimeTarget[] = { 5000, 7000, 10000 }; // bug 1: should be { ..., ...,
10000};
```

```
bool digInput5; // state of digital input 5
bool digInput6; // state of digital input 6
bool digInput7; // state of digital input 7
bool led; // state of led
int loopCounter;
// make variables of interest know to WawiLib:
void wawiVarDef()
{
    WawiSrv.wawiVar(digInput5);
    WawiSrv.wawiVar(digInput6);
    WawiSrv.wawiVar(digInput7);
    WawiSrv.wawiVar(led);
    WawiSrv.wawiVar(blinkTimeActual);
    WawiSrv.wawiVar(loopCounter);
    WawiSrv.wawiVarArray(blinkTimeTarget);
}
void setup()
{
    Serial.begin(115200);
    // initialize WawiLib library:
    WawiSrv.begin(wawiVarDef, Serial, "MyArduino");
    pinMode(LED, OUTPUT);
    pinMode(IN_5, INPUT);
    pinMode(IN_6, INPUT);
    pinMode(IN_7, INPUT);
    WawiSrv.wawiBreakDisable();
}
void loop()
{
    digInput5 = digitalRead(IN 5);
    digInput6 = digitalRead(IN 6);
    digInput7 = digitalRead(IN_7); // bug 2: should be digInput7 = ...
    if (digInput5)
        blinkTimeActual = blinkTimeTarget[0]; // bug 3: should be activeMsSetpoint[0]
    if (digInput6)
        blinkTimeActual = blinkTimeTarget[1];
    if (digInput7)
        blinkTimeActual = blinkTimeTarget[2];
    if (digInput5 || digInput6 || digInput7)
    {
        WawiSrv.wawiBreak(1, "breakpoint after write to activeMsCounter hit");
    }
    while (blinkTimeActual > 0) // bug 4: should be activeMsCounter > 0
    {
        WawiSrv.wawiBreak(2, "In while loop");
        WawiSrv.print("Counting down:");
        WawiSrv.println(blinkTimeActual);
        WawiSrv.println("LED is ON.");
        led = HIGH;
        digitalWrite(LED, led);
```

```
WawiSrv.delay(500);
blinkTimeActual = blinkTimeActual - 500;
WawiSrv.println("LED is OFF.");
led = LOW;
digitalWrite(LED, led);
WawiSrv.delay(500);
blinkTimeActual = blinkTimeActual - 500;
}
WawiSrv.loop();
loopCounter++;
```

Fig. 2.1. WawiRec source code.

The demo sketch contains the variables *delayOn* and *delayOff*. They determine the blinking timing of the LED. The variables *activeMsCounter* and *activeMsSetpoint* determine the time interval the LED will remain blinking. *DigInput5*, *digInput6* and *digInput7* mirror the state of the 3 digital inputs 5, 6 and 7. The function wawiVarDef shares the addresses of the variables with WawiLib.

If you have studied the WawiDebugUSB demo, you will be familiar with the rest of the program: *WawiServ.begin(wawiVarDef,Serial,"MyArduino")* initializes the library for communication over the USB port. *wawiVarDef()* is the function where addresses of the variables are shared. "MyArduino" is used as name reference for the board.

3. WawiLib data recording to disk file

3.1. Introduction

One of the properties of the Arduino environment is that it requires a bit more than basic knowledge to record data to your PC.

Programming data communication via serial, Ethernet, Wi-Fi and USB interfaces is not so easy within Windows. For sure if you want to do this the right way (overlapped I/O, non-blocking, multithreaded, automatic re-connect etc.).

Originally, WawiLib did not contain any data recording functions. As I wanted to create a PID controller and record the signals, I decided to add recording functions to the program. Typical applications where recording can be used are physics experiments at school or elsewhere.

Suppose you want to do some thermal experiments using a calorimeter. You can of course write down on paper the temperature of the calorimeter in time. But automated data recording creates additional value in this kind of experiment: increased accuracy, technical challenge and the notion of "big data". Not to forget that writing temperature values down on paper every 10 seconds is boring and outdated.

Connecting a DS18B20 high resolution temperature sensor to an Arduino board is not so challenging. But getting the measured data in an Excel table for further processing requires more specialized knowledge. With the data recording functions integrated in WawiLib, everybody will be able to create data recording files by configuring the right settings within WawiLib on the PC.

Mew	Copen	₽ Save	Print	Copy	X Cut	Daste	Offline	Setup()	Loop()	₩rite a	all Continue	e brkpt l	⊠ orkpt	
Avai	able variables		a transmission of the second s			Interface	/Ard. ID	Va	riable nan	ne	Actual value	Format	Recorder	Variab
🖻 se	r1\MyArduing	0		-		cor1/May	Arduino		dolayOn			INT	PECO	WAR FRR NOT FOUND
-	Single variabl	e				ser 1/My/	Arduino		delayOff			INT	REC2	VAR_ERR_NOT_FOUND
	digInput5			2		Sel 1/ Wy/	Arduno		delayOn			IINT	RECZ	VAR_ERR_NOT_FOUND
	digInput6			4		ser1/Mv/	Arduino	blin	kTimeActu	ıal	3000	INT	RFC1	@blinkTimeActual=0x0
				5		ser1/Mv/	Arduino	blink	TimeTarge	t[0]	5000	INT	REC1	@blinkTimeTarget=0x0
		ctual		6		ser1/Mv/	Arduino	blink	TimeTarge	t[1]	7000	INT	REC1	@blinkTimeTarget=0x0
	H loopCount	er		7		ser1/My/	Arduino	blink	TimeTarge	et[2]	10000	INT	REC1	@blinkTimeTarget=0x0
	Array			8										
	blinkTimeT	arget[02]	9		ser1/My/	Arduino		digInput5		1	INT	REC1	@digInput5=0x02A8 [1
	[4 byte]	x 3		1(D	ser1/My/	Arduino		digInput6		0	INT	REC1	@digInput6=0x02A7 [1
	-blinkTim	eTarget[0]	1	1	ser1/My/	Arduino		digInput7		0	INT	REC1	@digInput7=0x02A6 [1
	-blinkTim	eTarget[1]	12	2									
	blinkTim	eTarget[2]	13	3	ser1/My/	Arduino	lo	opCounte	r	0x0026	HEX	REC1	@loopCounter=0x029
				14	4									
				19	5	ser1/My/	Arduino		led		1	INT	REC1	@led=0x02A1 [1 byte]
				10	5									
				<	- 1									>
dex	Time		1	lode	Me	ssage								
nuex	mine			Joue	IVIC	ssage								

3.2. Data recorders

Fig. 3.1. Variables with multiple data recorders.

Data recording in WawiLib is implemented with a data recorder object. In the figure above, you see 11 variables. Data recorder REC1 will record the first 2 variables. Data recorder REC2 will record the last 9 variables. Many data recorders can work next to each other, each of them with their own file type and recording settings.

Every time a data recorder does a recording, it takes the actual value of all its variables from the PC memory and writes the values to a disk file or memory file in a single write job. Each recorder has its own data file. At the beginning of the line, the timestamp indicating the current PC time is added (details: see later).

The value of these variables in the PC will be more or less accurate depending on the speed used to refresh them. Refreshing is done by reading the values from the Arduino memory. There is a general refresh setting that can be overridden individually for each variable. To change general settings:

✓ In WawiLib, go to the menu "Settings/User preferences and license".



Fig. 3.2. Modify the default refresh interval of variable Arduino 🗇 WawiLib memory.

In the fig. 3.2., you see the default settings for the refresh interval of the variables read from the Arduino. If you make this value too small, the communication link with your Arduino will be overloaded. If you make it too large, there will be too much lag between the values recorded and the real values of the variables.

- ✓ Enter the value of 250 ms for the default refresh interval.
- ✓ Press "ok".

You can override the general timing for each individual variable (fig. 3.3):

✓ Click right on a variable and select "variable properties".

Dper	n Save	Print	Co	by Cu	nt Pas	ð te	Offline	Setup()	Loop()
	Interface/A	rd. ID	Variab	le name	Actual	value	Write	value	Format
	ser1		dela	ayOn					INT
	ser1	Va	ariable pr	operties	i				×
	ser1		Tincing						-
	ser1		liming	Recordin	ng				
			Serial /	ethernet	read				
	ser1		New ti	me base (msec):	500			1
	ser1								
	ser1								-
Tim	ne								_
				(OK	C	ancel		

Fig. 3.3. Modify the individual refresh interval of variable Arduino 🗇 WawiLib memory.

WawiLib can create 3 types of data recording files:

- xml: extensible Markup Language files
- xlsx: Microsoft Excel/Open Office compatible files
- csv: comma separated value files

Another topic is what to do with the recorded data when WawiLib is set offline and then reconnected by the user. Some users require that the existing data file is overwritten, others expect the new data to be appended to the data file and yet another might require a new file to be started.

le proper	tion D	IN IL DUL	1 64 4 44 44					
1.1	Record	de alls Disk	cusage and file size limita	tion				
Filename	and director	ry		Data file format				
Filename	e: WawiDat	aRecorded.c	sv	csv: comma se	eparated values			
Director	v: C:\Users\	Johi\Docum	nents		O xml: extensible markup language			
	,			xlsx: Excel/Libr	eOffice compatible	e spreadshee		
When go	oing online o	n Arduino:						
• Overv	write current	data file		CSV separator (\t=	tab) .			
Chippe	and new data	records to c	current data file					
O Start	with new data	records to a file (add st	current data file tart date and time to filen	ame)				
O Start	with new data	records to a file (add st	current data file tart date and time to filen	ame)				
Start	with new data	records to c	current data file tart date and time to filen	ame)				
O Start	with new data	records to c a file (add st	zurrent data file tart date and time to filen.	ame)				
Start	with new data	records to c	zurrent data file tart date and time to filen.	ame)				
O Start	with new data	records to c	zurrent data file tart date and time to filen.	ame)				
O Start	with new data	records to a file (add st	current data file	ame)	Addiesed by			
Configure	end new data with new dat ed data recore File mode	records to a a file (add st ders Time base	current data file tart date and time to filen File WawiDataRecorded cov	Dir	Add record typ	Add		
O Start O Start Configure Name F REC1 (end new data with new dat ed data record File mode OVERWRITE	records to a a file (add st ders Time base No	zurrent data file tart date and time to filen File WawiDataRecorded.csv	Dir C:\Users\Johi\Documents	Add record typ no	Add Remove		
Configure Name F REC1 (end new data with new dat ed data record File mode OVERWRITE	records to a a file (add st ders Time base No	zurrent data file tart date and time to filen File WawiDataRecorded.csv	ame) Dir C:\Users\Johi\Documents	Add record typ no	Add Remove Update		
Start	with new data	records to a a file (add st	zurrent data file tart date and time to filen.	ame)				
O Start O Start Configure Name F REC1 (end new data with new dat ed data record File mode OVERWRITE	records to a a file (add st ders Time base No	zurrent data file tart date and time to filen File WawiDataRecorded.csv	Dir C:\Users\Johi\Documents	Add record typ no	Add Remove		

Fig. 3.4. Data recorder settings with various options for file types, and file approach (append overwrite...).

All 3 options are available within WawiLib. If you choose to create a new file, the date and the time the file was created will be added to the data file name to distinguish one file from another. The name can be in UTC or local time. UTC is not impacted by winter time – summer time changes but local time depends on the area where you live.

3.3. Data recording trigger aspects

Different applications have different data trigger requirements. The calorimeter application from the previous section for example could require data recording every 10 seconds (time-based recording).

Another type of application would be registration of movement detection with a PIR infrared sensor module. The PIR output would be connected to an Arduino input. In this application, time-based registration is not the right way to go. This type of application requires an additional (time stamped) data record when the value of the output of the detector has changed. We are not interested in recording the PIR signal unless the output of the sensor changes (on-change recording).

- ✓ In WawiLib, go to "Settings/Data Recording".
- ✓ Select the tab "Record details".

ata recording s	ettings				;				
Recorder name:	REC1								
File properties	Record details	Disk usage and file si	ze limitation						
Recording trig	ggers ne based (for cha	nge based: modify pr	operties of variabl	e in main table)					
Recording int	Recording interval (sec): 1.0								
Data record ty									
Data record ti	mestamp setting	IS							
🗹 Add date		Add date (U	TC)						
🗹 Add time		🗌 Add time (U	TC)						
Add millis	seconds (approx	.) 🛛 🖂 Add relative	timestamp (elaps	ed time)					
Configured da	ta recorders Mode	Time base	File	Dir					
Nume	mode	nine buse	THC		Remove				
					Update				
۲				>	Clear list				
		ОК	Cancel	Default parameters					

Fig. 3.5. Modify the recording interval of a data recorder (WawiLib memory \Leftrightarrow Disk file).

If you check "Record time based", the data recorder will store all variable values each time the recording time interval has expired. The recording interval is specified in the "Recording interval (sec)" field. If you want to use on-change recording, you need to uncheck "Recording time based".

While using on-change recording, the variables will still be polled in the Arduino memory according to the communication timing settings. But the moment a variable change is detected in the PC, a trigger is sent to the data recorder to write the value of all the variables associated with this data recorder immediately to disk (or to memory for .xlsx files). Whether a variable change will trigger a write of all variables associated with the recorder or not can be defined in the properties section of each individual variable. You can access the properties of a variable by clicking right on the variable in the grid of the main window.

Typically you can use an integer that increments on each event, whenever its value increases, WawiLib sees this and writes all variables related to the current data recorder to dis.

The recorder does not wait for its recording time interval to elapse to trigger a write job. When the internal recorder timer elapses, data are written again to disk (if enabled). This function is very handy to store events when and if they happen.

The trigger to write data "on change" mentioned in the previous paragraph does not even have to be part of the variables written to a disk. It can act as a trigger that is not part of the recorded data. This function can also be configured in the properties section of the variable.

Each data record can be accompanied by a time stamp. The time stamp can contain the current date and time in local time or UTC (universal time coordinated) or both time stamps. An approximate time stamp in milliseconds and the number of seconds since the start of the data recording can also be added.

3.4. Data recording detailed example

3.6.1. Recorded variables

✓ Fill in the WawiLib table as in fig. 3.6. and press 'Setup()'.

没 Wa	awiLib-PC	[C:\Users\Jo	hi\Docu	iments\Nc	Name	e.Wif*]-[C:\Users\.	Iohi\Docur	nents\No!	Name.Wvf	*]					-		×
File Ed	dit Setting	gs Help													_			
2	B		8	13		X	<u> </u>	6	6	6	4	Þ	•	Ø				
New	Open	Save	Print	Cop	у	Cut	Paste	Offline	Setup()	Loop()	Write	all Continue	brkpt	brkpt				
⊟-Ava	ailable vari	ables					Interface	/Ard. ID	Vai	iable nan	ne	Actual value	Forma	at Re	corder			^
	Single	variable			1		ser1/My/	Arduino	blin	kTimeActu	Jal	0	INT	F	REC1	@blink1	imeActu	al=i
		Input5			2		ser1/My/	Arduino	blink	TimeTarge	et[0]	5000	INT	F	REC1	@blink1	imeTarg	et=
		Input6			3		ser1/My/	Arduino	blink	TimeTarge	et[1]	7000	INT	F	REC1	@blink1	imeTarg	et=
	⊞ dig	Input7			4		ser1/My/	Arduino	blink	TimeTarge	et[2]	10000	INT	F	REC1	@blink1	imeTarg	et=
	⊞-led				5													
	⊞-blir	hkTimeActua	al		6		ser1/My/	Arduino		digInput5		0	INT	F	REC1	@digIn	put5=0x0	32A
	⊞-loo	pCounter			7		ser1/My/	Arduino	C	digInput6		0	INT	F	REC1	@digIn	put6=0x0	D2A
	Array				8		ser1/My/	Arduino	¢	digInput7		0	INT	F	REC1	@digIn	put7=0x0	J2A
	🖨 blir	nkTimeTarge	et[02]		9													
		[4 byte] x 3			10		ser1/My/	Arduino	lo	opCounte	r	0x0608	HEX	F	REC1	@loop(Counter=	0x0
		blinkTimeTa	arget[0]		11													
		blinkTimeTa	arget[1]		12		ser1/My/	Arduino		led		0	INT	I	REC1	@led=0	x02A1 [1	by
		blinkTimeTa	arget[2]	<														>
Index	Time			Node	1	Mess	age											^
000	30/07/20	21 14:22:50	0.745	REC1		open	ing file C:\U	sers\Johi\D	ocuments	WawiDat	aRecord	led.csv for over	write OK					~
<							-											>
		Lo	pop()	Autowrite	on	REC	1 [RECO_W/	AIT_TRIG] c	nt=1 s	er <mark>1=MyA</mark> r	duino=0	COM18/115200	,8,N,1,AV	R [ITF_LC	OOP] ms	g.ok/tot	7183/7	183

Fig. 3.6. Variables linked to recorder 1.

- ✓ Disable and enable output window settings as in fig 3.6.
- ✓ (Enable "Display data recording". => What is written to disk during variable recording will be shown in the output window.)
- ✓ (Enable "Display output window recording". => What is written to disk during output recording will be shown in the output window.)
- ✓ (Enable "Automatic scroll" => The window will scroll to the last message added.)

	Display .print() messages	
	Display diagnostics messages	
	Display communication protocol messages	
\checkmark	Display data recording	
	Display output window recording	
	Copy selected text	Ctrl+C
	Clear Window	
\checkmark	Automatic scroll	
	Reset view	

Fig. 3.6. Enable display data recording in the output window & automatic scroll.

3.6.2. Time based recording of a variable or a series of variables

✓ Go to "Settings/Data Recording..."

corder n	ame: REC1									
le proper	rties Record	details Disk	usage and file size lim	itation						
Filenam Filenam Directo When g Oven O Appe	he and directo he: WawiRei ry: C:\Users going online o write current of end new data i	ry corded.csv \Johi\Docum n Arduino: data file records to cur	ents rent data file t date and time to file	Data file format CSV separator (\t:	separated values Ile markup languag breOffice compatib =tab) ;	je ble spreadsh	eet			
) Start	with new data	The (aud star								
) Start	with new data	lers								
Configure Name	with new data ed data recorc File mode	lers Time base	File	Dir	Add record type	Add date	Add time	Add date utc	Add tim	Add
Ostart Configure Name REC1	ed data record File mode OVERWRITE	lers Time base 10 sec	File WawiRecorded.csv	Dir C:\Users\Johi\Documents	Add record type no	Add date yes	Add time yes	Add date utc no	Add tim no	Add Remove Update

Fig. 3.7. Define REC1 as a data recorder time based.

✓ Fill in the table as indicated in fig. 3.7.

- ✓ Press "Add" and "Ok".
- ✓ Select tab 2.
- ✓ Select REC 1 in the list control.
- ✓ Change recording interval to 1.0 sec
- ✓ Press "Update"

Data recording settings				×						
Data recorder name: REC1										
File properties Record details	File properties Record details Disk usage and file size limitation									
Recording triggers ✓ Record time based (for c	change based: modify properties	of variable in main table)								
Recording interval (sec): 1.0										
Data record type	based/change based)									
Data record timestamp setti	ings									
✓ Add date	Add date (UTC)									
Add time	Add time (UTC)									
Add milliseconds (appro	ox.) 🗹 Add relative timestam	p (elapsed time)								
Configured data recorders										
Name File mode Time	base File	Dir	Add record typ	Add						
REC1 OVERWRITE 1 sec	WawiDataRecorded.csv	C:\Users\Johi\Documents	no	Remove						
				Remove						
				Update						
< Clear list										
OK Cancel Default parameters										

Fig. 3.8. Change time base of REC1 from 10 sec to 1 sec.

- ✓ Press OK.
- ✓ Press 'Setup ()'.
- ✓ Press "Offline".

In the output window, you see at line 003 the first record written to the data file. This is the title of the various columns (including the variable names). In the output window, you see what is written by Rec 1: delayOn and delayOff are both 500. On line 014 in the output window, you see the footer that is written when WawiLib goes offline.

- ✓ Start Excel or LibreOffice calc.
- ✓ Drag and drop the file (file name and location as indicated as indicated on line 014 above) in the Excel grid:

6	<u>_</u> ک		WawiDataReco	orded.c	sv	Sigr	n in	Ī]	—		×
F	ile Ho	me	nsert Page L Formu D	Data Re	view	iew Dev	elc	elp (ŢТ	ell me	R s∣	hare
E13	3	*	: × ✓ f	x								*
	В	С	D	E	F	G	Н	Т	J	К	L	
1	time	relative timestamp	blinkTimeActual (INT)	blinkTimeTarget (INT)	blinkTimeTarget[1-1] (INT)	blinkTimeTarget[2-2] (INT)	digInput5 (INT)	digInput6 (INT)	digInput7 (INT)	loopCounter (HEX)	led (INT)	
2	9:41:34	0	4500	5000	7000	10000	1	0	0	0x00CB	0	
3	9:41:35	1	3000	5000	7000	10000	1	0	0	0x00CB	1	
4	9:41:36	2	2500	5000	7000	10000	1	0	0	0x00CB	0	
5	9:41:37	3	1000	5000	7000	10000	1	0	0	0x00CB	1	
6	9:41:38	4	500	5000	7000	10000	1	0	0	0x00CB	0	
7	9:41:39	5	4000	5000	7000	10000	1	0	0	0x00CC	1	
8	9:41:40	6	3500	5000	7000	10000	1	0	0	0x00CC	0	
9	9:41:41	7	2500	5000	7000	10000	1	0	0	0x00CC	0	
10	9:41:41	7	file closed: offline									
11												
	•	Way	wiDataRecorded	+		1	•					
Read	dy Scroll	Lock	Display Setti	ngs						1	- + -	100%

Fig. 3.9. Data recorded opened in Microsoft Excel.

	WawiDataReco	orded.csv (r	ead-only) - LibreOffice	Calc			-		×			
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	Insert	F <u>o</u> rmat St <u>y</u> les <u>S</u> he	et <u>D</u> ata <u>T</u> ools <u>W</u> ind	ow <u>H</u> elp				⊕ ×			
	• 🚔 • 🜄	- 🗋 🕯		$ \not\leq \underline{A} \leftarrow \cdot \diamond \cdot$	Abc 📰 • 🔝 •	↑↓ ⊉↓ Z↓ 🖓 🛄 💵	🖉 Ω • 🛖					
E11	11 $\checkmark f_X \Sigma \bullet = $											
1	This document is open in read-only mode.											
	A	В	C	D	E	F	G	~	0.0			
1	date	time	relative timestamp	blinkTimeActual (INT)	blinkTimeTarget (INT)	blinkTimeTarget[1-1] (INT)	blinkTimeTarget[2-2]	<u>194</u>			
2	30/07/2021	9:41:34	0	4500	5000	7000			100			
3	30/07/2021	9:41:35	1	3000	5000	7000						
4	30/07/2021	9:41:36	2	2500	5000	7000						
5	30/07/2021	9:41:37	3	1000	5000	7000			5			
6	30/07/2021	9:41:38	4	500	5000	7000			Тx			
7	30/07/2021	9:41:39	5	4000	5000	7000						
8	30/07/2021	9:41:40	6	3500	5000	7000						
9	30/07/2021	9:41:41	7	2500	5000	7000		~				
<												
	WawiDataRecorded											
She	et 1 of 1			Default		🗈 🛛 Average: ; Sum: 0		-+	100%			

Fig. 3.10. Data recorded opened in LibreOffice Calc.

- ➡ If excel does not process the .csv file properly y can use the command Data/From Text/CSV to import the .csv formatted data.
- \Rightarrow Above you see the reslult of opening the file in LibreOffice Calc.

3.6.3. Change based recording of a variable or a series of variables

- ✓ Go to "Settings/Data Recording...".
- ✓ Press "Clear List".
- ✓ Fill in the fields as indicated in fig. 3.11:

Data recording setting	s						×
Data recorder name:	REC1						
File properties Record	details Disk	usage and	file size limitat	tion			
Filename and director Filename: WawiDa Directory: C:\Users When going online of Overwrite current Append new data Start with new da	vry taRecorded.xl \Johi\Docum on Arduino: data file a records to c ta file (add st	sx ents urrent data f art date and	file time to filena	ame)	Data file format Csv: comma se Xml: extensible Xsx: Excel/Libr CSV separator (\t=	eparated values e markup langu eOffice compa tab)	s iage tible spreadsheet
Configured data recor	rders Time base	File		Dir		Add record ty	/F Add
<							Remove Update Clear list
	(ЭK	Car	ncel	Default para	ameters	

Fig. 3.11. Creating a new data recorder, XLSX file format and overwrite mode.

- ✓ Select the tab "Record details" (fig. 3.12.).
- ✓ Unselect (disable) "Recording time based".

Data reco	ording settings							×				
Data recorder name: REC1												
File prop	erties Record	details Disk	usage and fi	le size <mark>limita</mark>	tion							
Recordin	ng triggers ord time based ing interval (se	d (for change ec): 1	e based: mod	ify propertie	s of variable in	main table)						
Data rec	cord type I record type (time based/o	change based)								
Data rec	cord timestam	p settings										
Add	d date		Add date	e (UTC)								
Add	d time		Add time	(UTC)								
Add	d milliseconds	(approx.)	Add relat	ive timestan	np (elapsed tim	e)			t			
Configu	red data recor	ders										
Name	File mode	Time base	File		Dir		Add record typ	Add				
REC1 OVERWRITE No WawiDataRecorded.csv C:\Users\Johi\Documents no Remov								Remove				
								Update				
<							N5' >	Clear list				
			OK	Car	ncel	Default para	meters					

Fig. 3.12. Creating a new data recorder, disable time interval based recording.

- ✓ Press "ADD".
- ✓ Press "OK" (close the dialog box).
- ✓ In the main variable grid, change the name of the recorders to REC2 as indicated below.
- ✓ In the main variable grid, select line 7 with the variable "*blinkTimeActual*". (fig. 3.13.)
- ✓ Right-click and open the menu "Variable properties".
- ✓ Go to Tab "Recording" in the dialog box.
- ✓ Check "Recording enabled" and "Change of variable triggers data recorder".

ser1/MyA	rduino	blinkTimeActu	0	INT	
ser1/MyA	Variable	properties		×	INT
ser1/MyA	and the second sec				INT
ser1/MyA	Timing	Recording			INT
ser1/MyA	Red	ording enabled			INT
ser1/MyA	Ch.	ange of variable trig	gers data reco	rder	INT
ser1/MyA	Defau	It value if read fails:	ERR		INT
ser1/MyA					HEX
ser1/MyA		ОК	Cantel		INT

Fig. 3.13. Make *blinkTimeActual* a trigger for variable change recording.

- ⇒ Note: If you disable (fig 3.13.) Recording enabled, this variable will trigger recording of the complete set of variables in REC1 but blinkActual will not be part of the recorded data.
- ✓ Enable "Trace data recording " in the output window.
- ✓ Press "OK" and Press 'Setup()'.
- ✓ On your Arduino board, connect Input 5 to 5V using the breadboard wire.
- ✓ Connect the input 5 6 and 7 to GND.
- ✓ Connect the input 5 to 5V.
- ✓ Connect the input 5 to GND.
- ✓ Look at the output window.
- ✓ Wait 10 seconds.
- ✓ Press "Offline".

INCOV	Dpen	₽ Save	🖶 Print	Copy	X Cut	🗈 Paste	Offline	Setup()	Loop()	Write a	Continue	• brkpt	⊠ brkpt	
- Ava	ilable variab	les				Interface/#	Ard. ID	Varia	ble name	e /	Actual value	Format	Recorder	
-	ser1\MyArd	uino		1		ser1/MvAr	duino	blinkT	imeActua	al	0	INT	REC1	@hlinkTime
	Single va	ariable		2		ser1/MvAr	duino	blinkTir	meTarget	[0]	5000	INT	REC1	@blinkTime
	digin	put5		3		ser1/MvAr	duino	blinkTir	meTarget	[1]	7000	INT	REC1	@blinkTime
	digInput6 distant			4		ser1/MvAr	duino	blinkTir	meTarget	[2]	10000	INT	REC1	@blinkTime
	⊞-led	put/		5										e sintine
	H-blink	TimeActu	al	6		ser1/MyAr	duino	dic	alnput5		0	INT	REC1	@diaInput5
		Counter		7		ser1/MyAr	duino	dig	gInput6		0	INT	REC1	@digInput6
	- Array			8		ser1/MyAr	duino	dig	gInput7		0	INT	REC1	@digInput7
	blink	TimeTarge	et[02]	9										
	-[4	byte] x 3		10		ser1/MyAr	duino	loop	Counter		0xEE61	HEX	REC1	@loopCour
	-bl	linkTimeTa	arget[0]	11										
	bl	linkTimeTa	arget[1]	12		ser1/MyAr	duino		led		0	INT	REC1	@led=0x02
	bl	linkTimeTa	arget[2]	13										
				<										,
Index	Time			Node	Mes	sage								
Index 000	Time 30/07/202*	1 14:32:34	4.167	Node REC1	Mes write	sage e footer: 30/0	07/2021 14	:32:34 583	3 file close	ed: offline	•			
Index 000 001	Time 30/07/202 ⁻ 30/07/202 ⁻	1 14:32:34 1 14:32:34	4.167 4.167	Node REC1 REC1	Mes write closi	sage e footer: 30/0 ng opened f)7/2021 14	l:32:34 583 s\Johi\Doc	3 file close uments\W	ed: offline VawiData	Recorded.csv	ОК		
Index 000 001 002	Time 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11	4.167 4.167 1.206	Node REC1 REC1 REC1 REC1	Mes write closi	sage footer: 30/0 ng opened f ning file C:\U)7/2021 14 ile C:\Users sers\Johi\[l:32:34 583 s\Johi\Doc Documents	3 file close uments\V \\WawiDat	ed: offline VawiData taRecorde	e Recorded.csv ed.csv for ove	OK rwrite OK		
Index 000 001 002 003	Time 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11	4.167 4.167 1.206 1.206	Node REC1 REC1 REC1 REC1 REC1	Mes write closi oper write	sage 9 footer: 30/0 ng opened f ning file C:\U 9 header: dat	07/2021 14 ile C:\Users sers\Johi\[e time rela	l:32:34 583 s\Johi\Doc Documents tive timesta	3 file close uments\W awiDat amp blink	ed: offline VawiData taRecorde cTimeActu	Recorded.csv ed.csv for ove ual (INT) blinkT	OK rwrite OK limeTarget	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004	Time 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11	4.167 4.167 1.206 1.206 1.206	Node REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <>	sage footer: 30/(ng opened f ning file C:\U header: dat (HEX) led (II)7/2021 14 ile C:\Users sers\Johi\[e time rela NT)	l:32:34 583 SJohi\Doc Documents tive timesta	3 file close uments\W \\WawiDat amp blink	ed: offline VawiData taRecorde TimeActu	e Recorded.csv (ed.csv for ove ual (INT) blink1	OK rwrite OK fimeTarget ((INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11	4.167 4.167 1.206 1.206 1.206 1.207	Node REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <> rec(o	sage e footer: 30/0 ng opened f ning file C:\U e header: dat (HEX) led (II o.c.): 30/07/2)7/2021 14 ile C:\Users sers\Johi\[e time rela NT) 021 14:49:	3:32:34 583 Sydohi\Doci Documents tive timesta 11 0 0 500	3 file close uments\W \WawiDat amp blink	ed: offline VawiData taRecord TimeActu 0000 0 0	e Recorded.csv ed.csv for ove ual (INT) blink1 0 0xB8DE 0	OK rwrite OK FimeTarget I	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21	4.167 4.167 1.206 1.206 1.206 1.207 1.379	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi ope write <> rec(o	sage e footer: 30/(ng opened f ning file C:\U e header: dat (HEX) led (II o.c.): 30/07/2 o.c.): 30/07/2)7/2021 14 ile C:\Users sers\Johi\E e time rela NT) 021 14:49: 021 14:49:	1:32:34 583 3:\Johi\Doc Documents tive timesta 11 0 0 500 21 9 5000	3 file close uments\W WawiDat amp blink 00 7000 10 5000 700	ed: offline VawiData taRecorde TimeActu 0000 0 0 00 10000	Recorded.csv (ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2	OK rwrite OK FimeTarget I	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:21 1 14:49:21	4.167 4.167 1.206 1.206 1.206 1.207 1.379 1.994	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <> rec(o rec(o	sage footer: 30/(ng opened f ning file C:\U header: dat (HEX) led (II o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\E e time rela VT) 021 14:49: 021 14:49: 021 14:49:	1:32:34 583 (Contemports) Cocuments (Cocuments) (Cocum	3 file close uments\W \WawiDat amp blink 00 7000 10 5000 700 5000 700	ed: offline VawiData taRecorde TimeActu 0000 0 0 00 10000 00 10000	Recorded.csv (ed.csv for ove aal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2	OK rwrite OK TimeTarget I 1 0	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22	4.167 4.167 1.206 1.206 1.206 1.207 1.379 1.994 2.258	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <> rec(o rec(o rec(o	sage e footer: 30/0 ng opened f ning file C:\U e header: date (HEX) led (If o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\E e time relat NT) 021 14:49: 021 14:49: 021 14:49: 021 14:49:	1:32:34 583 (Johi\Doc Occuments tive timesta 11 0 0 500 21 9 5000 22 9 4500 22 10 4000	3 file close uments\W \WawiDat amp blink 00 7000 1(5000 700 5000 700 0 5000 700	ed: offline VawiData taRecorde (TimeActu 0000 0 0 00 10000 00 10000	Recorded.csv / ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK fimeTarget 1 1 0 2 1	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 005 006 007 008 009	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:22	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi ope write <> rec(o rec(o rec(o	sage e footer: 30/0 ng opened f hing file C:\U e header: date (HEX) led (II o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time relat NT) 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49:	3:32:34 583 (x) Ohi\Doc Documents tive timesta 11 0 0 500 21 9 5000 22 9 4500 22 10 4000 22 10 4000 23 10 3500	3 file close uments\W \WawiDat amp blink 00 7000 10 5000 700 5000 700 0 5000 70 0 5000 70	ed: offline VawiData taRecorde (TimeActu 0000 0 0 00 10000 00 10000 000 10000	Recorded.csv (ed.csv for ove aal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK fimeTarget 1 0 2 1 2 0	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 009 010	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:22 1 14:49:22	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi ope write <> rec(o rec(o rec(o rec(o	sage e footer: 30/0 ng opened f hing file C:\U e header: dat (HEX) led (II o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time relai NT) 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49:	3:32:34 583 (Johi)Doc Documents tive timesta 11 0 0 500 21 9 5000 22 9 4500 22 10 4000 23 10 3500 23 11 3000	3 file close uments\W \WawiDat amp blink 5000 700 5000 700 0 5000 70 0 5000 70 0 5000 70 0 5000 70	ed: offline VawiData taRecorde (TimeActu 0000 0 0 00 10000 00 10000 000 10000 000 10000	Recorded.csv (ed.csv for ove aal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2 0 1 0 0 0x5EB2 0 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK fimeTarget f 0 2 1 2 0 2 1	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 010 011	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:22 1 14:49:23 1 14:49:24	4.167 4.167 1.206 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes vrite closi ope write <> rec(o rec(o rec(o rec(o	sage a footer: 30/0 ng opened f ning file C:UU a header: dat (HEX) led (If o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time relai NT) 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49:	3:32:34 583 (Johi)Doc Documents tive timesta 11 0 0 500 21 9 5000 22 9 4500 22 10 4000 23 10 3500 23 11 3000 24 12 2500	3 file close uments\W amp blink 00 7000 10 5000 700 5000 700 0 5000 70 0 5000 70 0 5000 70 0 5000 70 0 5000 70	ed: offline VawiData taRecorde (TimeActor) 00000000 0010000 00010000 00010000 00010000	Recorded.csv (ed.csv for ove aal (INT) blinkT 0 0x88DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK fimeTarget f 0 2 1 2 0 2 1 2 0 2 1 2 0	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 010 011 012	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:22 1 14:49:22 1 14:49:24 1 14:49:24	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005 4.264	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <> rec(o rec(o rec(o rec(o rec(o	sage a footer: 30/0 ng opened f hing file C:UU a header: dat (HEX) led (If o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time rela NT) 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49: 021 14:49:	:32:34 583 (c) (c) (c) (c) (c) (c) (c) (c) (c) (c)	3 file close uments\W WawiDat amp blink 00 7000 10 5000 700 0 5000 700 0 5000 70 0 5000 70 0 5000 70 0 5000 70 0 5000 70	ed: offline VawiData taRecords TimeActu 0000 0 0 00 10000 000 10000 000 10000 000 10000 000 10000	e Recorded.csv f ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK TimeTarget 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 010 011 012 013	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:22 1 14:49:24 1 14:49:24 1 14:49:24	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005 4.264 4.826	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <> rec(o rec(o rec(o rec(o rec(o	sage a footer: 30/0 ng opened f ning file C:UU a header: dat (HEX) led (If o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time rela NT) 021 14:49: 021 14:49	1:32:34 58:3 (3 file close uments\W \WawiDat amp blink 00 7000 10 5000 700 0 5000 700 0 5000 70 0 5000 70 0 5000 70 0 5000 70 0 5000 70 0 5000 70	ed: offline VawiData taRecorde TimeActu 0000 0 0 00 10000 000 10000 000 10000 000 10000 000 10000 000 10000	e Recorded.csv f ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK TimeTarget 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:24 1 14:49:24 1 14:49:24 1 14:49:25	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005 4.264 4.826 5.442	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closi oper write <> rec(o rec(o rec(o rec(o rec(o rec(o	a footer: 30/0 ng opened f hing file C:UU a header: dat (HEX) led (II b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2 b.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time rela NT) 021 14:49: 021 14:49	132:34 583 (c) (c) (c) (c) (c) (c) (c) (c) (c) (c)	3 file close uments\W \WawiDat amp blink 00 7000 1(5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 70 0 5000 70 0 5000 70 0 5000 70 0 5000 70	ed: offline VawiData taRecord (TimeActu 0000 0 0 00 10000 000 10000 000 10000 000 10000 000 10000 000 10000 000 10000	e Recorded.csv f ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2	OK rwrite OK fiimeTarget 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:24 1 14:49:24 1 14:49:25 1 14:49:26	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005 4.264 4.826 5.442 5.009	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mess write closi open write <> rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c) rec(c rec(c) rec(c	a footer: 30/0 ng opened f hing file C:UU a header: dat (HEX) led (II o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time rela NT) 021 14:49: 021 14:49	132:34 583 (c) (c) (c) (c) (c) (c) (c) (c) (c) (c)	3 file close uments\W \WawiDat amp blink 00 7000 10 5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 700 5000 700	ed: offline VawiData taRecord (TimeActu 0000 0 0 00 10000 00 10000 000 10000 000 10000 000 10000 000 10000 000 10000	e Recorded.csv f ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2 1 0 0 0x5EB2	OK rwrite OK fimeTarget 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 0	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 000 000 000 000 010 010 011 012 013 014 015 016	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:24 1 14:49:24 1 14:49:26 1 14:49:26 1 14:49:26	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005 4.264 4.826 5.442 5.009 5.274	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closis open write <> rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c) re	a footer: 30/0 ng opened f ning file C:UU a header: dat (HEX) led (II o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time rela VT) 021 14:49: 021 14:49	132:34 583 (c) (c) (c) (c) (c) (c) (c) (c) (c) (c)	3 file close uments\W \\WawiDat amp blink 0 7000 10 5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 700 5000 700 5000 700	ed: offline VawiData taRecord (TimeActu 0000 0 0 00 10000 000 10000 000 10000 000 10000 000 10000 000 10000 000 10000 00 10000 00 10000	e Recorded.csv f ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 0 0x629F 0	OK rwrite OK fiimeTarget 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 0 0	(INT) blinkTim	eTarget[1-1]
Index 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016	Time 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202 30/07/202	1 14:32:34 1 14:32:34 1 14:49:11 1 14:49:11 1 14:49:11 1 14:49:21 1 14:49:22 1 14:49:22 1 14:49:22 1 14:49:24 1 14:49:25 1 14:49:26 1 14:49:26	4.167 4.167 1.206 1.206 1.207 1.379 1.994 2.258 2.828 3.440 4.005 4.264 4.826 5.442 5.009 5.274	Node REC1 REC1 REC1 REC1 REC1 REC1 REC1 REC1	Mes write closis oper write <> rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c rec(c) re	sage a footer: 30/C ng opened f ning file C:\U a header: dat: (HEX) led (If o.c.): 30/07/2 o.c.): 30/07/2	07/2021 14 ile C:\Users sers\Johi\[e time rela VT) 021 14:49: 021 14:49	1:32:34 583 (c) (c) (c) (c) (c) (c) (c) (c) (c) (c)	3 file close uments\W \WawiDat amp blink 0 7000 10 5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 700 0 5000 700 5000 700 0 5000 700	ed: offline VawiData taRecord (TimeActu 0000 0 0 00 10000 00 10000 000 10000 000 10000 000 10000 000 10000 000 10000 000 10000 000 10000	e Recorded.csv f ed.csv for ove Jal (INT) blinkT 0 0xB8DE 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 1 0 0 0x5EB2 1 0 0 0x5EB2 0 0 0x629F 0	OK rwrite OK fiimeTarget 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 2 0 2 1 0	(INT) blinkTim	eTarget[1-1]

Fig. 3.13. On change recording example result.

On each alternation of *blinkTimeActual*, a new data set will be written to the data file. Each write operation is also visible in the output window. Once *blinkTimeActual* no longer changes, the recording stops.

✓ Open the file (see line 002 of the output window) in Excel or LibreOffice calc:

	' 5 ~ ⊘ ~	Ŧ			WawiD	ataRecordeo	d csv - Exce	I	Sig	n in 🖸	J –		×
File	Home	Insert F	Page Layout	Formula	s Data	Review	View	Developer	Help	Q Tell m	ne	$\mathcal{A}_{\!$	are
A1		X	f _x	date									*
	А	В	С	D	Е	F	G	Н	1	J	К	L	
1 da	te	time	relative tir	blinkTime	blinkTime	blinkTime	blinkTime	digInput5	digInput6	digInput7	loopCour	led (INT)	
2	30/07/2021	14:49:11	0	0	5000	7000	10000	0	0	0	OxB8DE	C)
3	30/07/2021	14:49:21	9	5000	5000	7000	10000	1	0	0	0x5EB2	1	L
4	30/07/2021	14:49:22	9	4500	5000	7000	10000	1	0	0	0x5EB2	C)
5	30/07/2021	14:49:22	10	4000	5000	7000	10000	1	0	0	0x5EB2	1	L
6	30/07/2021	14:49:23	10	3500	5000	7000	10000	1	0	0	0x5EB2	C)
7	30/07/2021	14:49:23	11	3000	5000	7000	10000	1	0	0	0x5EB2	1	L
8	30/07/2021	14:49:24	12	2500	5000	7000	10000	1	0	0	0x5EB2	C)
9	30/07/2021	14:49:24	12	2000	5000	7000	10000	1	0	0	Ox5EB2	1	[
10	30/07/2021	14:49:25	12	1500	5000	7000	10000	1	0	0	0x5EB2	C)
11	30/07/2021	14:49:25	13	1000	5000	7000	10000	1	0	0	Ox5EB2	1	1
12	30/07/2021	14:49:26	14	500	5000	7000	10000	1	0	0	0x5EB2	C)
13	30/07/2021	14:49:26	14	0	5000	7000	10000	0	0	0	0x629F	C)
14	30/07/2021	14:50:31	78	file closed	offline								
15	WawiD	ataRecor	ded (+)				: •					•
Ready	Scroll Lock	3				1	🔓 Display Se	ttings		巴	- I-		00%

Fig. 3.14. On change recording example result.

In the table above, you can see that the status changes of *blinkTimeAcutal* are the only ones to trigger the addition of a new data record to the data recording file. Other variables are recorded at the same moment. As *digInput5* goes to 1, *blinkTimeAcutal* starts decreasing. Do note that digInput5 remains recorded as 1 because it is not updated as long as the CPU is executing the while() loop.

If we would enable "Recording time based" for recorder REC2, it would record data both time based and change based. Typically, you could decide to record at a slow pace time based and use "on change" to have updated data when a trigger comes indicating something interesting happened.

3.5. Data recorder storage aspects

One of the biggest challenges of data recording is how to manage your disk space. Imagine a weather station application. It runs day in and day out, recording its data in a single file. This would be not so handy because the file risks to become so large that it cannot be handled any more.

In the same sense there are limits to the amount of space a program can occupy on your hard disk. You do not want to get into trouble with other programs because WawiLib recorded data are eating up too much of your free disk space.

In order to tackle these 2 issues, every data recorder in WawiLib is able to delete its old data files if the amount of disk space used by the data recorder is too large.

Old data files can only be deleted once they are closed. The disk cleanup is triggered each time the data recorder closes a data file. So you need to restart with a new file regularly.

Data recordir	ng settings					×					
Recorder nar	ne: REC2										
File properti	es Record details	Disk usage	and file size limitation								
Filename a	and directory			Data file format							
Filename:	WawiRecorded.	xlsx		🔾 csv: comma separa	ated values						
Directory:	C:\Users\Johi\D	ocuments		 xml: extensible ma xlsx: Excel/LibreOf 	rkup langua <u>c</u> fice compatil	ge ble spreadsheet					
When goir	ng online on Ardui	no:		CSV separator (\t=tab)	;						
Overwrite current data file											
Start wi	Start with new data file (add start date and time to filename)										
Configured	data recorders										
Name	Mode	Time base	File	Dir	Add recor	Add					
REC2	TIMESTAMPED	None	WawiRecorded.xlsx	C:\Users\Johi\Documents	no						
						Remove					
						Update					
						Clear list					
<					>						
		ОК	Cancel	Default parame	ters						

Fig. 3.15. Data recorder that creates a new data file each 15 minutes, hour, or day.

In the data recorder settings on tab 3 ("Disk usage and file size limitation") you can enable the function to limit the amount of disk space used by the current data recorder. Files that have the same name as indicated in the "Filename tab" will be deleted, the oldest one first. This option is only valid for recorders that create new files every 15 minutes, every hour or every day.

arecording	g settings									
order name	ne: REC2									
e properties	s Record d	letails Disk	usage and file size limi	itation						
Disk usage										
Limit disl	sk usage for	this recorder	(delete oldest files)							
Max disk us	sage for this	recorder (M	B):		10					
tart with ne	ew datafile	(filename = o	late + time)							
) Never										
Each 15 m	minutes									
) Each hour	Jr									
) Each day Protect aga	ainst data lo	oss (power fai	ilure)	If date/time are	part of filename	() - 10				
) Each day Protect aga Close file onfigured d	ainst data lo ile after each data recorde	oss (power fai n write opera ers	ilure) tion (slows down reco	If date/time are rding). Use UTC in f	part of filename ile names instead o	f local time.				
) Each day Protect aga Close fil onfigured d Name File	, ainst data lo ile after each data recordo ile mode	oss (power fai n write opera ers Time base	ilure) tion (slows down reco File	rding). If date/time are	part of filename ile names instead o Add record type	f local time. Add date	Add time	Add date utc	Add tin	Add
) Each day Protect aga Close fil onfigured d Name File REC2 OV	ainst data lo ile after each data recorde ile mode WERWRITE	oss (power fai h write opera ers Time base No	llure) tion (slows down reco File WawiRecorded.xlsx	If date/time are If date/time are Use UTC in f Dir C:\Users\Johi\Documents	part of filename ile names instead o Add record type no	f local time. Add date yes	Add time yes	Add date utc no	Add tin no	Add
) Each day Protect aga Close fil onfigured d Name File REC2 OV	, ainst data lo ile after each data recorde ile mode WERWRITE	oss (power fai n write opera ers Time base No	llure) tion (slows down reco File WawiRecorded.xlsx	If date/time are If use UTC in f Use UTC in f Dir C:\Users\Johi\Documents	part of filename ile names instead o Add record type no	f local time. Add date yes	Add time yes	Add date utc no	Add tin no	Add Remove
) Each day Protect aga Close fil onfigured d Name Fil REC2 OV	ainst data lo ile after each data recorde ile mode WERWRITE	oss (power fai h write opera ers Time base No	llure) tion (slows down reco File WawiRecorded.xlsx	If date/time are Use UTC in f Dir C:\Users\Johi\Documents	part of filename ile names instead o Add record type no	f local time. Add date yes	Add time yes	Add date utc no	Add tin no	Add Remove Update
) Each day Protect aga Close fil onfigured d Name Fila REC2 OV	, ile after each data recordd ile mode WERWRITE	n write opera n write opera ers Time base No	ilure) tion (slows down reco File WawiRecorded.xlsx	If date/time are If date/time are Use UTC in f Dir C:\Users\Johi\Documents	part of filename ile names instead o Add record type no	f local time. Add date yes	Add time yes	Add date utc no	Add tin no	Add Remove Update Clear list
) Each day Protect aga Close fil onfigured d Name Fil REC2 OV	, ainst data lo ile after each data recordd ile mode IVERWRITE	n write opera n write opera ers Time base No	llure) tion (slows down reco File WawiRecorded.xlsx	If date/time are Use UTC in f Dir C:\Users\Johi\Documents	part of filename ile names instead o Add record type no	f local time. Add date yes	Add time yes	Add date utc no	Add tin no	Add Remove Update Clear list

Fig. 3.16. Data recorder that creates a new data file each 15 minutes.

Above you see the check box "Close file after each write operation. This option makes sure your data ends up on disk immediately. In case of the weather station application, recording data each 15 minutes is more than fast enough. If we close the file after each write, we are much less vulnerable to failures of the grid feeding our computer and other anomalies.

4. WawiLib .print() recording to disk file.

4.1. Introduction

One of the properties of the Arduino environment is that it requires a bit more than basic knowledge to record data to your PC.

Programming data communication via serial, Ethernet, Wi-Fi and USB interfaces is not so easy within Windows. For sure if you want to do this the right way (overlapped I/O, non-blocking, multithreaded, automatic re-connect etc.).

In many applications one needs the ability to register output of a program during a prolonged period of time. Typically this can be an application that contains a very difficult to find bug or this can be an application where the user needs to register events that happen from time to time.

Originally, WawiLib did not contain any output recording functions. From time to time Sylvester Solutions does provide consultancy services and in one case the user wanted to register continuously the output of the sketch on a remote server via Wi-Fi. This is where the idea to create output recorders very similar to the data recorder in the previous chapters originated.

4.2. Define an output recorder

- ✓ Go to "Settings/Data Recording...".
- ✓ Press "Clear List".

utput recor Output reco	ding settin order name	gs : ORI	:C1					>
le propertie File name a Filename: Directory: When goin Overwri Appenc Start wi	s Output ind directo WawiOuti C:\Users\ g online or te current of new data th new data	messag ry Johi\Do n Ardui data file record a file (a	ge recording detail orded.csv ocuments no: e s to current data fil dd start date and t	e e ime to filename)	e size limitation Data file fo © csv: coi O xml: exi O xlsx: Exi CSV separa			
Configured	output rec	orders						Add
Name File	e mode Fi	e Dir	Add record type	Add Ard. WawiSrv.	orintf() output	Add Ard. diag msg	Add Ard. protoco	Remove Update Clear list
<							>	
			ОК	C	ancel	Default parameters	;	

Fig 4.1. define an output recorder (file type)

✓ Fill in the fields as indicated in fig. 4.1

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	
File properties Output message recording details Disk usage and file size limitation Arduino boards/interfaces to record Date record timestamp settings	
Date record timestamp settings Add date Add time (UTC) Add time Add relative timestamp (elapsed time) Add milliseconds (approx.) Add date (UTC) 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	
✓ Add date ☐ Add time (UTC) ✓ Add time ☐ Add relative timestamp (elapsed time) ☐ Add milliseconds (approx.) ☐ Add date (UTC) Add source and record type to file	
Image: Add relative timestamp (elapsed time) Add milliseconds (approx.) Add date (UTC) 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) Image: Arduino "WawiSrv.print()" messages Image: Disagnostics messages Image: Disagnostics messages	
Add milliseconds (approx.) Add date (UTC) Add source and record type to file	
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) Arduino "WawiSrv.print()" 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()	7 laa
OREC1 OVERWRITE WawiOutputRecorded.csv C:\Users\Johi\Documents no yes	kemove
	Update
C	Clear list
< >	
OK Cancel Default parameters	

Fig 4.2. select the data to be recorded with the output recorder.

- ✓ Go to tap 2
- ✓ Fill in the fields as in fig. 4.2.
- ✓ Press "Add"
- ✓ Press "OK"
- ✓ Press "Setup" in the main tool bar.
- ✓ Enable the output window Diagnostic message display as in Fig. 4.3.



- ✓ Fig 4.. Display settings of the output window.
- ✓ Connect digital input 5,6,7 to GND.
- ✓ Connect digital input 5 to 5V.
- ✓ Connect digital input 5 to GND.

👸 Wa	wiLib-PC [C	:\Users\Jo	hi\Docur	nent	s\NoNar	me.Wif*]-[C:\Users\J	ohi∖Docur	nents\No	Name.Wvf*]				- 🗆 X
File Ed	dit Settings	Help			-		-	-		100000	2		12	1.00	
New	Dpen 🖻	Save	Print		Сору	اللا Cut	Paste	Offline	Setup()	Loop()	₩rite	all Continue	• brkpt	e brkpt	
- Ava	iilable variat	oles		^			Interface,	Ard. ID	Va	riable nam	e	Actual value	Forma	t Recorder	Varia ^
B -	ser1\MyArc	luino			1		ser1/MvA	Arduino	blin	kTimeActu	al	0	INT	RFC1	@blinkTimeActual=0x02A2 [4 byte] x 1
	Single v	ariable			2		ser1/MvA	Arduino	blink	TimeTarge	:[0]	5000	INT	REC1	@blinkTimeTarget=0x0100 [4 byte] x 3
	digir	iput5			3		ser1/MvA	Arduino	blinkTimeTarget[1]		7000	INT	REC1	<pre>@blinkTimeTarget=0x0100 [4 byte] x 3</pre>	
		iputo			4		ser1/MvA	rduino	blinkTimeTarget[2]		10000	INT	REC1	<pre>@blinkTimeTarget=0x0100 [4 byte] x 3</pre>	
		iput/			5				Dimenneralget[2]					Gommerninger oxoroo [royte] x o	
	- blink	TimeActu	al		6		ser1/MyArduino		diaInput5		0	INT	REC1	@diaInput5=0x02A8 [1 byte] x 1 VAR	
		Counter	ui -		7		ser1/MyArduino			digInput6		0	INT	REC1	@digInput6=0x02A7 [1 byte] x 1 VAR
	Array				8		ser1/MyA	Arduino		digInput7		0	INT	REC1	@digInput7=0x02A6 [1 byte] x 1 VAR_
	=- blink	TimeTarge	et[02]		9										
	-[4	4 byte] x 3			10)	ser1/MyA	Arduino	lo	opCounter		0x62CF	HEX	REC1	@loopCounter=0x029F [2 byte] x 1 VA 🗸
	b	linkTimeTa	arget[0]	~	<										>
Index	Time			N	ode	Mess	age								^
004	30/07/202	1 14:07:48	3.306	RE	C1	ERRC	R opening f	ile C:\User	s\Johi\Do	cuments\W	awiDat	aRecorded.xlsx	for overwi	ite: The proces	s cannot access the file because it is being
005	30/07/202	1 14:10:35	5.995	0	REC1	write	footer:								
006	30/07/202	1 14:10:36	5.047	0	REC1	closin	ig opened fi	le C:\Users	\Johi\Doc	uments\W	awiOut	putRecorded.csv	OK		
007	30/07/202	1 14:10:58	3.957	0	REC1	open	ing file C:\U	sers\Johi\E	ocument	s\WawiOut	putReco	orded.csv for ov	erwrite Ol	(
008	30/07/202	1 14:10:58	3.957	0	REC1	write	header: date	e time							
009	30/07/202	1 14:11:00	0.269	R	C1	open	ing file C:\U	sers\Johi\[ocument	s\WawiData	Record	led.xlsx for over	write OK		
010	30/07/202	1 14:11:00	0.270	RE	C1	write	header: date	e time relat	tive timest	amp blink	imeAct	ual (INT) blinkTi	meTarget	(INT) blinkTime	Target[1-1] (INT) blinkTimeTarget[2-2] (INT)
011	30/07/202	1 14:11:00	0.270	RE	C1	<>	(HEX) led (IN	IT)							
012	30/07/202	1 14:11:00	0.270	RE	C1	rec(o.	c.): 30/07/20	021 14:11:	00 0 0 500	00 7000 10	000 0 0	0 0 0x19CF 0			
013	30/07/202	1 14:11:09	9.518	0	REC1	rec. o	utput: 30/01	7/2021 14:	11:09 Cou	unting dow	n:5000				
014	30/07/202	1 14:11:09	9.518	0	REC1	rec. o	utput: 30/01	7/2021 14:	11:09 LED	is ON.					
015	30/07/202	1 14:11:09	9.565	R	C1	rec(o.	c.): 30/07/20	021 14:11:	10 8 5000	5000 7000	10000	1 0 0 0x7BC2 1			
016	30/07/202	1 14:11:10	0.024	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:10 LED	is OFF.					
017	30/07/202	1 14:11:10	0.134	RE	C1	rec(o.	c.): 30/07/20	021 14:11:	10 9 4500	5000 7000	10000	1 0 0 0x7BC2 0)		
018	30/07/202	1 14:11:10	0.531	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:10 Cou	unting dow	n:4000				
019	30/07/202	1 14:11:10	0.531	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:10 LED	is ON.					
020	30/07/202	1 14:11:10	0.696	R	C1	rec(o.	c.): 30/07/20	021 14:11:	11 9 4000	5000 7000	10000	0 1 0 0 0x7BC2 1			
021	30/07/202	1 14:11:11	1.039	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:11 LED	is OFF.					
022	30/07/202	1 14:11:11	1.263	RE	C1	rec(o.	c.): 30/07/20	021 14:11:	11 10 350	0 5000 700	00 1000	00 1 0 0 0x7BC2	0		
023	30/07/202	1 <mark>14:11:</mark> 11	1.547	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:12 Cou	unting dow	n:3000				
024	30/07/202	1 14:11:11	1.547	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:12 LED	is ON.					
025	30/07/202	1 14:11:11	1.581	R	C1	rec(o.	c.): 30/07/20	021 14:11:	12 10 300	0 5000 700	00 1000	00 1 0 0 0x7BC2	1		
026	30/07/202	1 14:11:12	2.055	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:12 LED	is OFF.					
027	30/07/202	1 14:11:12	2.150	RE	EC1	rec(o.	.c.): 30/07/20	021 14:11:	12 11 250	0 5000 700	00 1000	00 1 0 0 0x7BC2	0		
028	30/07/202	1 14:11:12	2.560	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:13 Cou	unting dow	n:2000				
029	30/07/202	1 14:11:12	2.560	0	REC1	rec. o	utput: 30/07	7/2021 14:	11:13 LED	is ON.					
030	30/07/202	1 14:11:12	2.710	RE	C1	rec(o.	c.): 30/07/20	021 14:11:	13 11 200	0 5000 700	00 1000	00 1 0 0 0x7BC2	1		
031	30/07/202	1 14:11:13	3.018	0	REC1	rec. o	utput: 30/0	//2021 14:	11:13 LED	is OFF.					
032	30/07/202	1 14:11:1:	3.274	RE	:C1	rec(o.	c.): 30/07/20	J21 14:11:	13 12 150	0 5000 700	1000	00 1 0 0 0x/BC2	0		
033	30/07/202	1 14:11:1:	3.525	0	RECT	rec. o	utput: 30/01	//2021 14:	11:14 Cou	inting dow	n:1000				
034	30/07/202	1 14:11:13	3.525	0	RECT	rec. o	utput: 30/01	/2021 14:	11:14 LED	IS ON.					
035	30/07/202	1 14:11:13	0.039	RE		rec(o.	c.j: 30/07/20	JZ1 14:11:	14 12 100	U 5000 700	1000	0 1 0 0 0x/BC2	1		
030	30/07/202	1 14:11:14	+.034	0		rec. o	utput: 30/0	72021 14:	11:14 LED	IS UFF.	10000	1000-7000	N N		
037	20/07/202	1 14:11:14	+.133	Kt	C1	rec(o.	c.j: 30/07/20	JZT 14:11:	14 13 500	000 7000 4	0000 0	0.0.0.000000000	,		
020	20/07/202	1 14:11:14	1./ 10	R	C1	rec(o.	footor: 20/07/20	7/2021 14:11:	13 13 0 50	file closed	offlin	0 0 0X0A00 0			
040	20/07/202	1 14.11:28	2214	Rt Dr		Save	Cilleore lab	Docum	ntc\\Wavel	DataPacere	lod vler	ok			
040	20/07/202	1 14:11:20	210	RE	DEC1	Save	footor:	indocume	ints/waWil	Datanecord	ed.xisx	UK.			
041	30/07/202	1 14.11:20	2 202	0	RECI	docin	opened f		VION Des		awiOut	putRecorded	OK		
<	50/07/202	1 14.11:28		0	NEC I	CIOSIF	ig opened fi	ie C. (Users		aments\W	awiOut	puthecolded.CS\			
					C	ffline	Autowrite	on ORI	EC1 [RECO	_IDLE] cnt=	15	ser1=MyArduin	o=COM18	/115200,8,N,1	AVR [ITF_OFFLINE] msg.ok/tot: 7128/7128

Fig 4.4. Output recording working with Display of messages in output window (facultative).

- ✓ Wait 10 seconds.
- ✓ Press "Offline".
- \checkmark Open the file with recorded output data (see bottom line in the output window fig 4.4)

```
WawiOutputRecorded.csv - Notepad
                                              X
File Edit Format View Help
date;time;
30/07/2021;14:11:09;Counting down:5000;
30/07/2021;14:11:09;LED is ON.;
30/07/2021;14:11:10;LED is OFF.;
30/07/2021;14:11:10;Counting down:4000;
30/07/2021;14:11:10;LED is ON.;
30/07/2021;14:11:11;LED is OFF.;
30/07/2021;14:11:12;Counting down:3000;
30/07/2021;14:11:12;LED is ON.;
30/07/2021;14:11:12;LED is OFF.;
30/07/2021;14:11:13;Counting down:2000;
30/07/2021;14:11:13;LED is ON.;
30/07/2021;14:11:13;LED is OFF.;
30/07/2021;14:11:14;Counting down:1000;
30/07/2021;14:11:14;LED is ON.;
30/07/2021;14:11:14;LED is OFF.;
30/07/2021;14:11:28;End recording.;
<
Ln 1, Col 1
                  100%
                         Windows (CRLF)
                                         UTF-8
```

Fig 4.5. Output of .print() messages in the sketch recorded in a disk file.

Note: the data in the file was created with the statements in fig. 4.6. marked in yellow.

```
/*
* Project Name: WawiRecUsb
 File: WawiRecUsb.ino
* Detailed manual:
 www.SylvesterSolutions.com\documentation -> "Recording variables with WawiLib.pdf"
* Description: demo file library for WawiSerialUsb library.
* Data recorder demo.
* => Record values of variables to disk
* => Record .print() output to disk
* Use the USB programming port to make connection with the Arduino board.
* Variables can be checked & modified with the WawiLib-PC software.
* Author: John Gijs.
* Created March 2020
* More info: www.sylvestersolutions.com
* Technical support: support@sylvestersolutions.com
* Additional info: info@sylvestersolutions.com
*/
#include <WawiSerialUsb.h>
WawiSerialUsb WawiSrv;
#define LED 13 // blinking light
#define IN_5 5 // light start blinking switch 1
```

```
#define IN_6 6 // light start blinking switch 2
#define IN_7 7 // light start blinking switch 3
// variables for demo:
long int blinkTimeActual = 0; // counter blink active (milliseconds)
long int blinkTimeTarget[] = { 5000, 7000, 10000 }; // bug 1: should be { ..., ...,
10000;
bool digInput5; // state of digital input 5
bool digInput6; // state of digital input 6
bool digInput7; // state of digital input 7
bool led; // state of led
int loopCounter;
// make variables of interest know to WawiLib:
void wawiVarDef()
{
    WawiSrv.wawiVar(digInput5);
    WawiSrv.wawiVar(digInput6);
    WawiSrv.wawiVar(digInput7);
    WawiSrv.wawiVar(led);
    WawiSrv.wawiVar(blinkTimeActual);
    WawiSrv.wawiVar(loopCounter);
    WawiSrv.wawiVarArray(blinkTimeTarget);
}
void setup()
{
    Serial.begin(115200);
    // initialize WawiLib library:
    WawiSrv.begin(wawiVarDef, Serial, "MyArduino");
    pinMode(LED, OUTPUT);
    pinMode(IN_5, INPUT);
    pinMode(IN_6, INPUT);
    pinMode(IN 7, INPUT);
    WawiSrv.wawiBreakDisable();
}
void loop()
{
    digInput5 = digitalRead(IN_5);
    digInput6 = digitalRead(IN_6);
    digInput7 = digitalRead(IN_7); // bug 2: should be digInput7 = ...
    if (digInput5)
        blinkTimeActual = blinkTimeTarget[0]; // bug 3: should be activeMsSetpoint[0]
    if (digInput6)
        blinkTimeActual = blinkTimeTarget[1];
    if (digInput7)
        blinkTimeActual = blinkTimeTarget[2];
    if (digInput5 || digInput6 || digInput7)
    {
        WawiSrv.wawiBreak(1, "breakpoint after write to activeMsCounter hit");
    }
    while (blinkTimeActual > 0) // bug 4: should be activeMsCounter > 0
    {
```

```
WawiSrv.wawiBreak(2, "In while loop");
   WawiSrv.print("Counting down:");
   WawiSrv.println(blinkTimeActual);
   WawiSrv.println("LED is ON.");
    led = HIGH;
    digitalWrite(LED, led);
   WawiSrv.delay(500);
    blinkTimeActual = blinkTimeActual - 500;
   WawiSrv.println("LED is OFF.");
    led = LOW;
    digitalWrite(LED, led);
   WawiSrv.delay(500);
    blinkTimeActual = blinkTimeActual - 500;
}
WawiSrv.loop();
loopCounter++;
```

Fig 4.5. WawiRecUpb creating output to be recorded on a disk file.

5. Final notes

An important aspect is the use of PC memory for temporary storage of recorded data by the data recorders.

If you choose .xlsx as a data format, all the recorded data is stored into PC memory until the recording is ended. At that time a series of files is written to disk. An .xlsx file is in fact a zipped combination of xml coded files. The files contain data and references in xml format. So, storage in memory is an aspect that is linked to the concept of .xlsx files, there is no other way.

.csv files and .xml file recording works differently: data is written to the disk memory cache each time an additional record of data is recorded. This means that at a failure of the power to the PC, you will lose some records but a part of the .csv file could remain intact. For .xml files, data is written to the disk cache as well but in case of abnormal termination, the final closing records of the .xml file will not be added, so the file becomes corrupt. (You can try to recover a corrupted xml file using a text editor)

In order to minimize the risk of data loss, you can decide to create a new file every 15 minutes to make sure this data is saved to disk in case of power failure.

Another relevant aspect of data storage is the fact that recorders will start only if all of their tags can be read. When a recorder starts, it waits until all its tags are read once before it starts to record. This means that in case of an illegal/missing tag configuration, the recorder will continue to wait. You can observe the status of the data recorders in the bottom line of WawiLib-PC:

REC_WAIT_FOR_TAG_READ_ONCE is the state of the recorder used to wait for successful read of all tags.

The best way to check if a recorder is started properly is to activate data recorder tracing and to look at the output window. If recorded data is appearing in the output window, it is working properly.

6. Further reading

This demo demonstrates how to record variables with WawiLib. Recording can be done time-based or on-change. Data files can have .csv or .xlsx format. File sizes can be limited by automatically restarting with a new file every 15 minutes, every hour or every day. Disk usage can be limited for each datalogger. WawiLib can clean up its data files automatically to prevent disk space usage overload.

This demo also shows how to record the output of your .print() statements used in a sketch. Also there the output of the statements can be sent to files of different types and sizes.

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