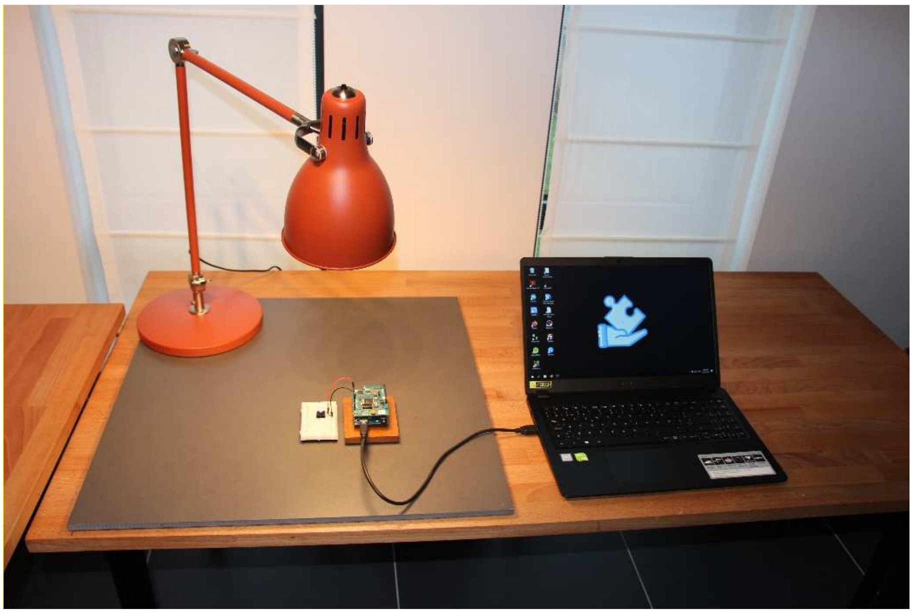
Debugging with WawiLib



1. Introduction

1.1. Objective of this document.

The objective of this document is to describe how to use WawiLib to debug an Arduino application.

WawiBlinkDebugUsb.ino, a demo sketch supplied with the WawiSerialUsb library, will be used to explain the concept.

The idea to make this demo came when I was making a demo for the WawiLib data recorder functions. The data recorder demo did not work so I used WawiLib to debug the demo. I was very surprised to see how fast I found the bugs with WawiLib.

In this document, I will describe what went wrong and how I used WawiLib to solve the problem. The demo is not an artificial setup but a report of what actually happened.

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.

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 but other boards can be used in a similar or even identical way. For compatibility of boards, go to www.sylvestersolutions.com.

If you want to use another Arduino board instead of the UNO for the demo, you might need to use 3V instead of 5V in order not to damage your board. For more details, look at the specification of your board.

1.3. Required user experience

The concepts of this document build further on the tutorial "*Getting started WawiLib USB*". You should be familiar with the demo as I will not re-explain how to go online with the board over USB.

2. THE “WawiBlinkDebugUSB” Demo sketch example

2.1. Concept of “WawiBlinkDebugUSB”

This application builds further on “WawiBlink” from the demo “Getting Started with WawiLib USB”. WawiBlinkDebugUSB uses 3 digital inputs. Whenever one of these inputs goes high, the onboard LED will blink during a time interval and then go dark again. The objective of the sketch is to blink a LED multiple times during a time period when one of 3 digital inputs is raised high.

- On digital input 5 high, the LED should blink during 3 seconds (500ms on, 500ms off).
- On digital input 6 high, the LED should blink during 7 seconds (500ms on, 500ms off).
- On digital input 7 high, the LED should blink during 10 seconds (500ms on, 500ms off).

In order to demonstrate how to debug with Wawilib, I added 4 bugs to the code. In this demo I will demonstrate how they can be found in a simple way using WawiLib.

Note: WawiLib does not claim to be a state-of-the-art debugger like dedicated IDE’s that use special hardware and dedicated CPU facilities. But, as will be demonstrated in this demo, using WawiLib together with the Arduino IDE will make it much easier to find bugs compared to what you can do with the bare Arduino-IDE.

2.2. Download and execute “WawiBlinkDebugUSB”

⇨ Open the example via the menu “File\Examples\WawiSerialUsb\WawiBlinkDebugUsb” in the Arduino IDE.

⇨ Compile and download the example in your board.

Below you can see the program in detail. This program will not work properly because there is are 4 bugs in it. We will use WawiLib to find the error and then correct it. The bugs are indicated in comment.

Compile and download WawiBlinkDebugUsb to your Arduino board:

```
/*
  Project Name: WawiBlinkDebugUsb
  File: WawiBlinkDebugUsb.ino

  Detailed manual:
  www.SylvesterSolutions.com\documentation -> "Debugging with WawiLib.pdf"

  Description: demo file library for WawiSerialUsb libary.
  Demo dedicated to demonstrate a debug concept with WawiLib.
  Lets you monitor and modify variables of different type and sizes.
  Use the programming USB port to make connection with the Arduino board.
  Variables can be checked & modified with the WawiLib-PC software.

  Author: John Gijs.
  Created December 2020
  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[] = { 3000, 7000, 1000 }; // bug 1: { ..., ..., 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);
    digInput6 = digitalRead(IN_7); // bug 2: should be digInput7 = ...


    if (digInput5)
        blinkTimeActual = blinkTimeTarget[1]; // bug 3: should blinktimeTarget[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 blinkTimeActual > 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. WawiBlinkDebugUsb.ino including 4 bugs.

⇨ Using 3 Dupont wires, connect all IO's 5,6, & 7 to the GND of the UNO.

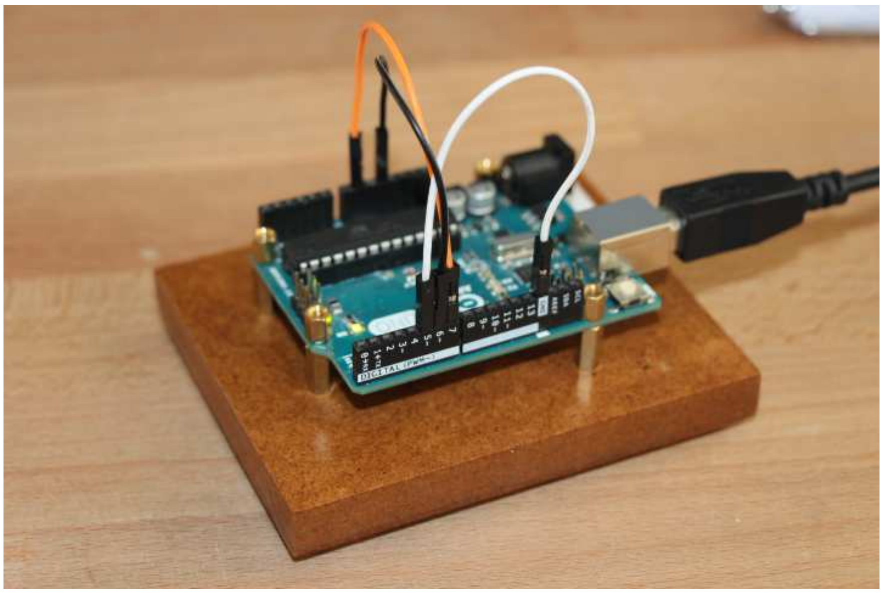
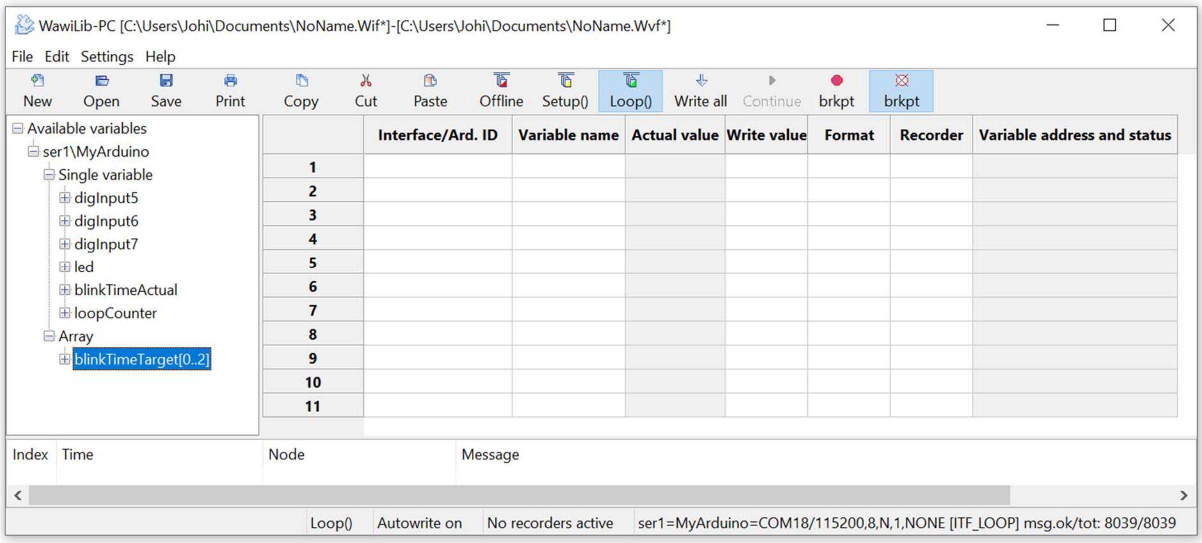


Fig. 2.2. Uno with IO's 5, 6, 7 connected to GND.

3. Debug with WawiLib

3.1. Visualize all sketch variables in WawiLib.

⇨ Start WawiLib on your PC.
⇨ Go online (press Setup()) on the top toolbar.


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt

Available variables
- ser1\MyArduino
  - Single variable
    - digInput5
    - digInput6
    - digInput7
    - led
    - blinkTimeActual
    - loopCounter
  - Array
    - blinkTimeTarget[0..2]

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Recorder	Variable address and status
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							

Index	Time	Node	Message

Loop() | Autowrite on | No recorders active | ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 8039/8039

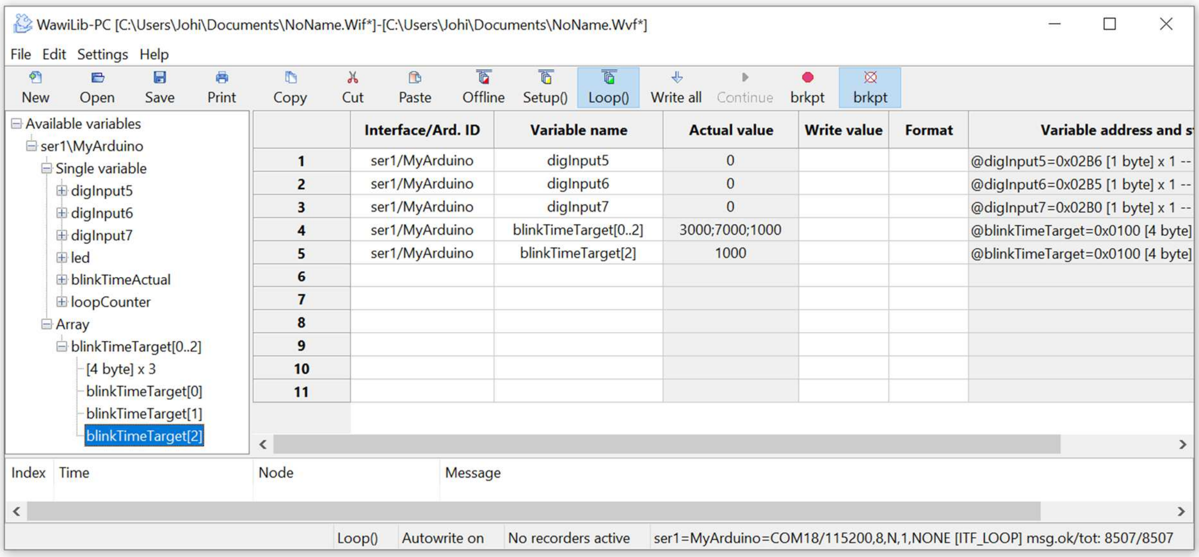
Fig. 3.1. WawiLib online on an Uno with WawiBlinkDebugUsb.ino

⇨ Drag the variables *digiInput5*, *digiInput6* and *digiInput7* to the grid.
⇨ Drag the variable *blinkTimeTarget[0..2]* to the grid (or type in the name of the variable in the table manually).
⇨ Note: you can select all of them together and drag all variables at once to the grid.

3.2. Bug 1: Faulty initialization of *blinkITemTarget*. (Check initializations)

⇨ Open the the blinkTimeTarget in the tree.

⇨ Add *blinkTimeTarget*[2] to the grid table.



WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Variable address and s
1	ser1/MyArduino	digInput5	0			@digInput5=0x02B6 [1 byte] x 1 --
2	ser1/MyArduino	digInput6	0			@digInput6=0x02B5 [1 byte] x 1 --
3	ser1/MyArduino	digInput7	0			@digInput7=0x02B0 [1 byte] x 1 --
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;1000			@blinkTimeTarget=0x0100 [4 byte]
5	ser1/MyArduino	blinkTimeTarget[2]	1000			@blinkTimeTarget=0x0100 [4 byte]
6						
7						
8						
9						
10						
11						

Index	Time	Node	Message

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 8507/8507

Fig. 3.2. Actual values of variable real time displayed by WawiLib.

⇨ Look at the values of *blinkTimeTarget [2]*

⇨ You see that they do not have the right value, 1000 instead of 10000.

✓ Look at the code in the Arduino IDE: the initialization value is faulty:

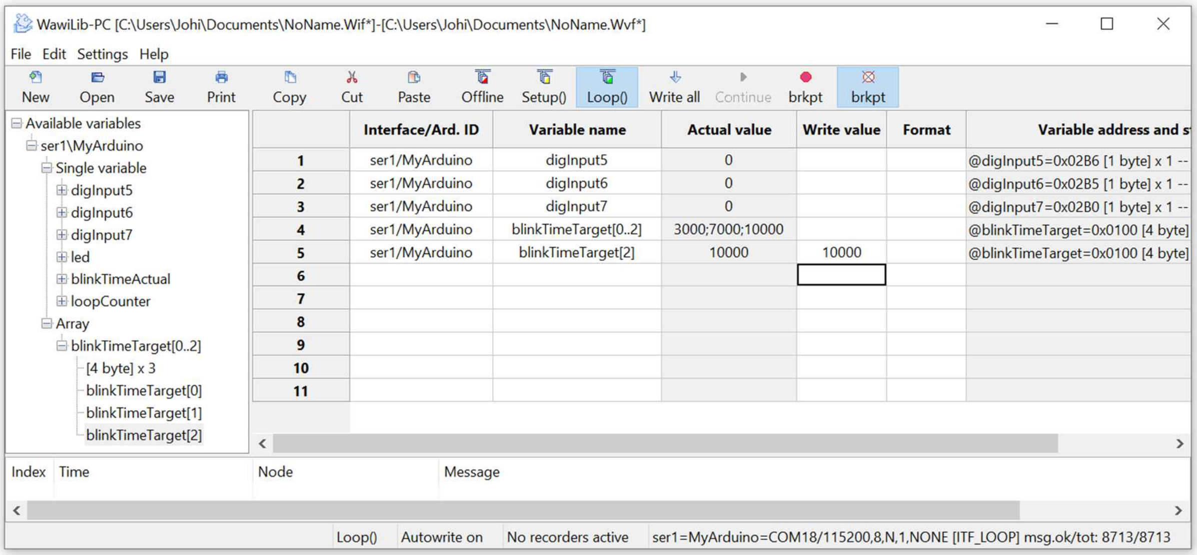
```
⇨ long int blinkTimeTarget[] = { 3000, 7000, 1000 }; // bug 1: { ..., ..., 10000};
```

Fig. 3.3. Bug 1.

✓ Correct the bug in the code (replace 1000 by 10000).

✓ Fill in 10.000 for the new value of *blinkTimeTarget[2]* in the "Write value" column.

✓ Press "write all".



WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Variable address and s
1	ser1/MyArduino	digInput5	0			@digInput5=0x02B6 [1 byte] x 1 --
2	ser1/MyArduino	digInput6	0			@digInput6=0x02B5 [1 byte] x 1 --
3	ser1/MyArduino	digInput7	0			@digInput7=0x02B0 [1 byte] x 1 --
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000			@blinkTimeTarget=0x0100 [4 byte]
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000		@blinkTimeTarget=0x0100 [4 byte]
6						
7						
8						
9						
10						
11						

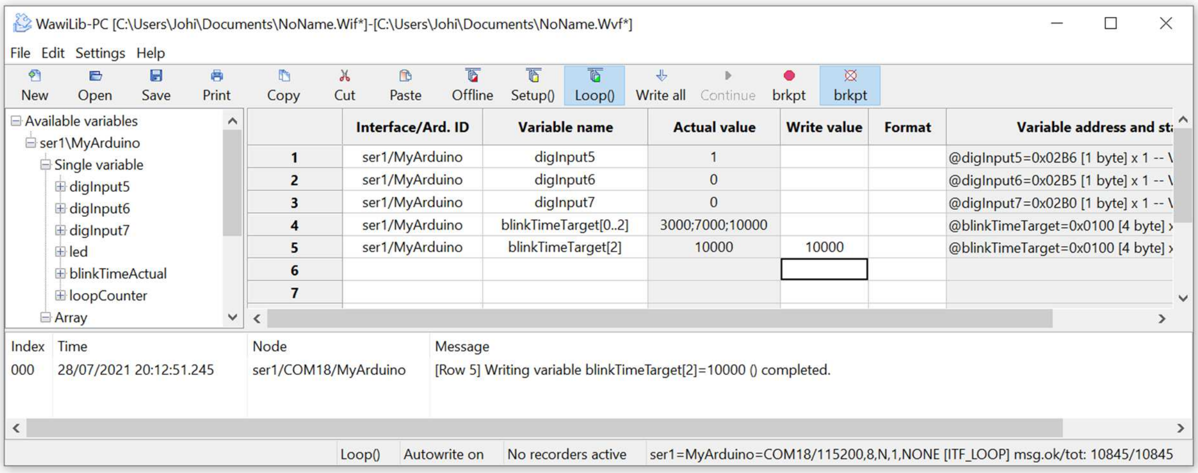
Index	Time	Node	Message

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 8713/8713

Fig. 3.4. Actual values after correcting the value of *blinkTimeTarget[2]* to 10000.

3.3. Bug 2: Use of *digiInput6*. Instead of *digiInput7*. (The IO test.)

⇨ Connect Digital io 5 to +5V on the Uno.
⇨ Look at the table.


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

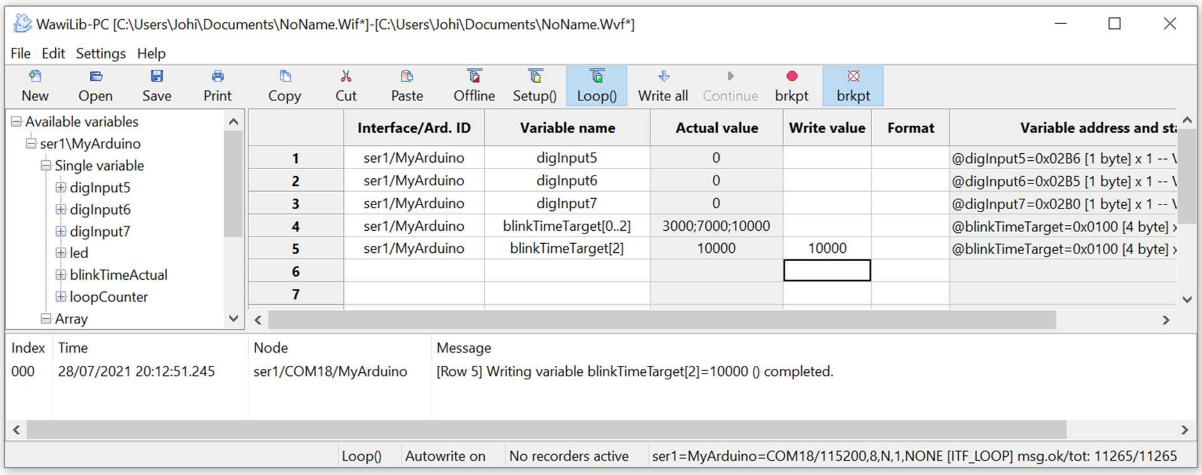
	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Variable address and st
1	ser1/MyArduino	digInput5	1			@digInput5=0x02B6 [1 byte] x 1 -- \
2	ser1/MyArduino	digInput6	0			@digInput6=0x02B5 [1 byte] x 1 -- \
3	ser1/MyArduino	digInput7	0			@digInput7=0x02B0 [1 byte] x 1 -- \
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000			@blinkTimeTarget=0x0100 [4 byte] ›
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000		@blinkTimeTarget=0x0100 [4 byte] ›
6						
7						

Index	Time	Node	Message
000	28/07/2021 20:12:51.245	ser1/COM18/MyArduino	[Row 5] Writing variable blinkTimeTarget[2]=10000 () completed.

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 10845/10845

Fig. 3.5. Actual values after DI5 connected to 5V.

⇨ *digInput5* goes to 1 as it should, this is OK.
✓ Connect Digital input 5 to GND on the Uno.
✓ Connect Digital input 6 to 5V on the Uno.
⇨ Look at the table


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Variable address and st
1	ser1/MyArduino	digInput5	0			@digInput5=0x02B6 [1 byte] x 1 -- \
2	ser1/MyArduino	digInput6	0			@digInput6=0x02B5 [1 byte] x 1 -- \
3	ser1/MyArduino	digInput7	0			@digInput7=0x02B0 [1 byte] x 1 -- \
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000			@blinkTimeTarget=0x0100 [4 byte] ›
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000		@blinkTimeTarget=0x0100 [4 byte] ›
6						
7						

Index	Time	Node	Message
000	28/07/2021 20:12:51.245	ser1/COM18/MyArduino	[Row 5] Writing variable blinkTimeTarget[2]=10000 () completed.

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 11265/11265

Fig. 3.6. Actual values after DI6 connected to 5V.

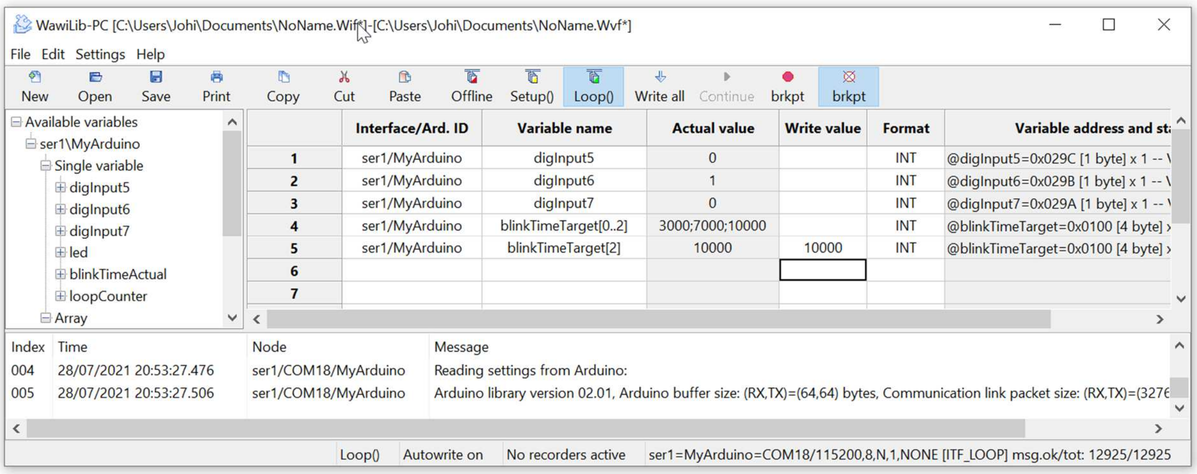
⇨ *digInput6* remains at 0, this is not OK, this is a bug. It seems that digInput6 is overwritten by the statement `digInput6 = digitalRead(IN_7);`
✓ Correct the bug in the code (fig. 3.7.).

```
    digInput5 = digitalRead(IN_5);
    digInput6 = digitalRead(IN_6);
    digInput7 = digitalRead(IN_7); // bug 2: should be digInput7 = ...
```

Fig. 3.7. Bug 2.

✓ Press "Offline"

- ✓ Compile and download the corrected code.
- ✓ Press "Setup".


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

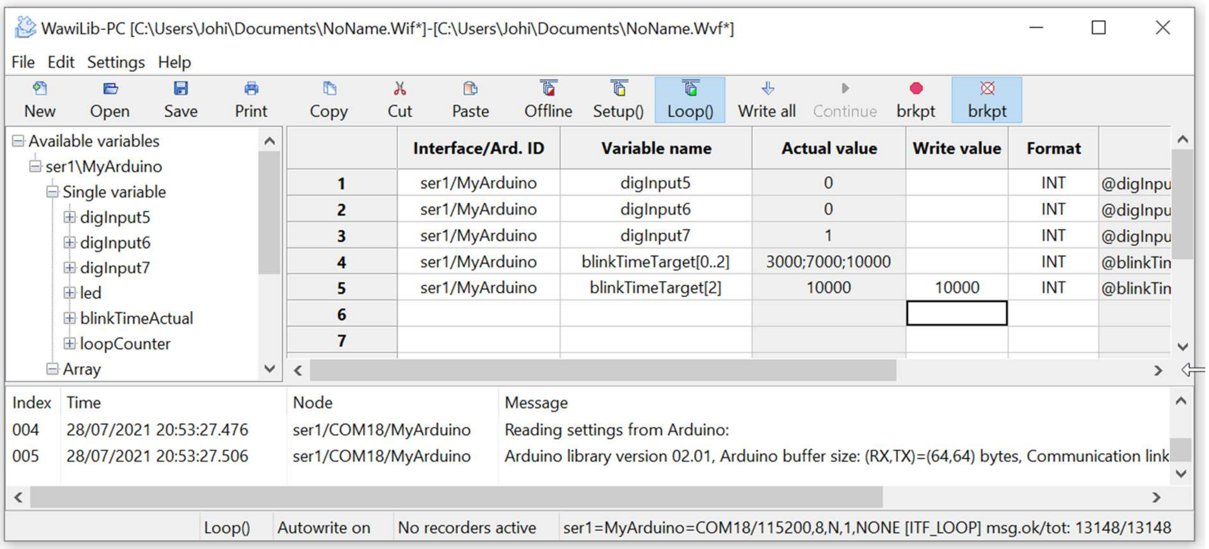
	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Variable address and st
1	ser1/MyArduino	digInput5	0		INT	@digInput5=0x029C [1 byte] x 1 -- \
2	ser1/MyArduino	digInput6	1		INT	@digInput6=0x029B [1 byte] x 1 -- \
3	ser1/MyArduino	digInput7	0		INT	@digInput7=0x029A [1 byte] x 1 -- \
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTimeTarget=0x0100 [4 byte] ›
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTimeTarget=0x0100 [4 byte] ›
6						
7						

```
Index  Time                     Node                    Message
004    28/07/2021 20:53:27.476  ser1/COM18/MyArduino    Reading settings from Arduino:
005    28/07/2021 20:53:27.506  ser1/COM18/MyArduino    Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link packet size: (RX,TX)=(3276
```

```
Loop()  Autowrite on  No recorders active  ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 12925/12925
```

Fig. 3.8. DI 6 at 5V makes digInput6 = 1 (OK).

- ✓ Connect Digital input 5 to GND.
- ✓ Connect Digital input 7 to 5V.
- ✓ Check the table:


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	0		INT	@digInpu
2	ser1/MyArduino	digInput6	0		INT	@digInpu
3	ser1/MyArduino	digInput7	1		INT	@digInpu
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTin
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTin
6						
7						

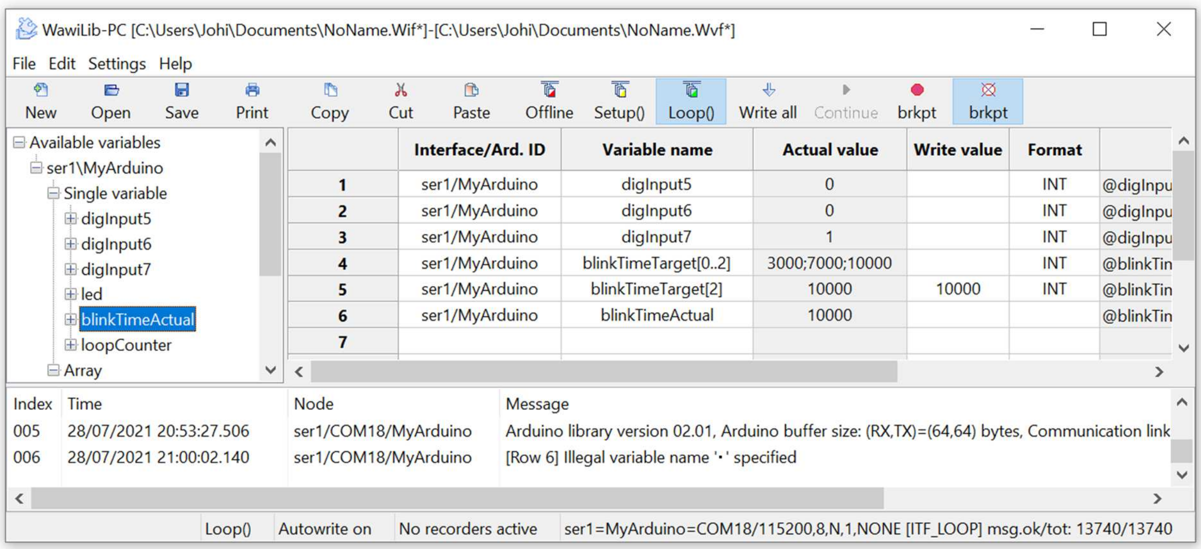
```
Index  Time                     Node                    Message
004    28/07/2021 20:53:27.476  ser1/COM18/MyArduino    Reading settings from Arduino:
005    28/07/2021 20:53:27.506  ser1/COM18/MyArduino    Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link
```

```
Loop()  Autowrite on  No recorders active  ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 13148/13148
```

Fig. 3.9. DI 7 at 5V makes digInput7 = 1 (OK).

3.4. Bug 3 blinkTimeTarget[1] should be blinktimeTarget[0].

- ✓ Add *blinkTimeActual* to the table.
- ✓ Connect Digital input 5 to GND.
- ✓ Connect Digital input 6 to GND.
- ✓ Connect Digital input 7 to 5V.
- ✓ Check the value of blinkTimeActual (should be 10.000)


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	0		INT	@digInpu
2	ser1/MyArduino	digInput6	0		INT	@digInpu
3	ser1/MyArduino	digInput7	1		INT	@digInpu
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTin
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTin
6	ser1/MyArduino	blinkTimeActual	10000			@blinkTin
7						

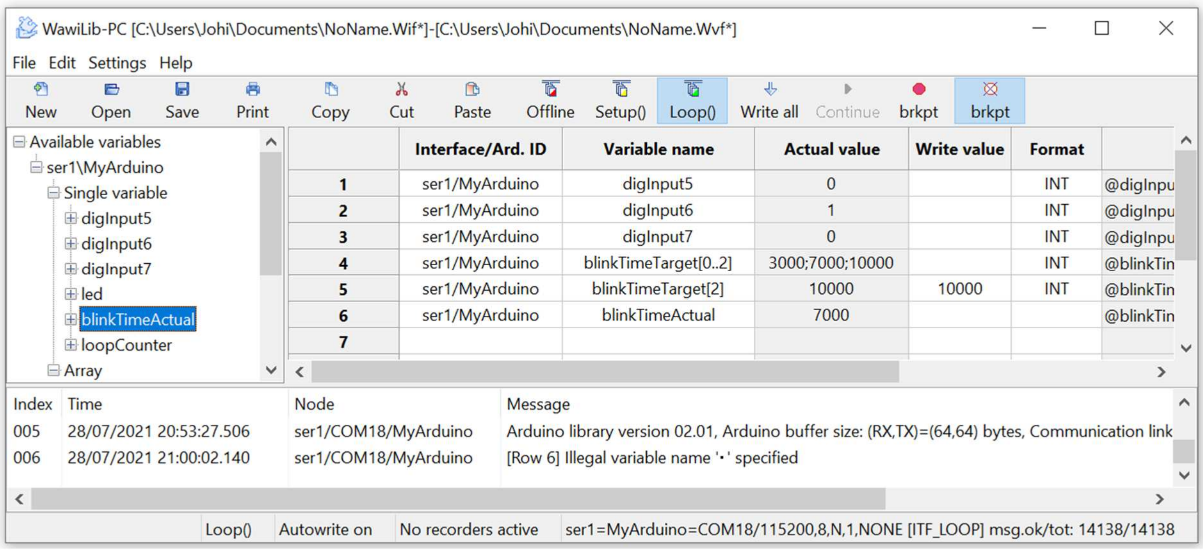
Index	Time	Node	Message
005	28/07/2021 20:53:27.506	ser1/COM18/MyArduino	Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link
006	28/07/2021 21:00:02.140	ser1/COM18/MyArduino	[Row 6] Illegal variable name '•' specified

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 13740/13740

Fig. 3.9. DI 7 at 5V makes blinkTimeActual 10.000 (OK).

⇨ Check the value of blinkTimeActual is and should be 10.000 (= OK)

- ✓ Connect Digital input 5 to GND.
- ✓ Connect Digital input 6 to 5V.
- ✓ Connect Digital input 7 to GND.
- ✓ Check the value of blinkTimeActual (should be 7.000)


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	0		INT	@digInpu
2	ser1/MyArduino	digInput6	1		INT	@digInpu
3	ser1/MyArduino	digInput7	0		INT	@digInpu
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTin
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTin
6	ser1/MyArduino	blinkTimeActual	7000			@blinkTin
7						

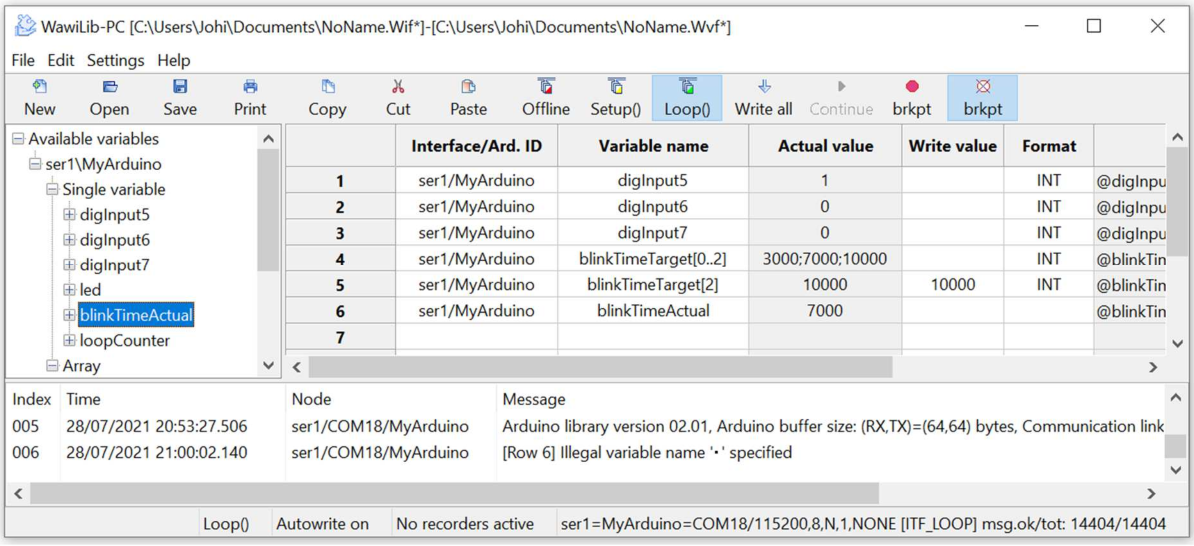
Index	Time	Node	Message
005	28/07/2021 20:53:27.506	ser1/COM18/MyArduino	Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link
006	28/07/2021 21:00:02.140	ser1/COM18/MyArduino	[Row 6] Illegal variable name '•' specified

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 14138/14138

Fig. 3.10. DI 6 at 5V makes blinkTimeActual 7.000 (OK).

⇨ Check the value of blinkTimeActual is and should be 7.000 (= OK)

- ✓ Connect Digital input 5 to 5V.
- ✓ Connect Digital input 6 to GND.
- ✓ Connect Digital input 7 to GND.
- ✓ Check the value of blinkTimeActual (should be 3.000)



WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	1		INT	@digInpu
2	ser1/MyArduino	digInput6	0		INT	@digInpu
3	ser1/MyArduino	digInput7	0		INT	@digInpu
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTin
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTin
6	ser1/MyArduino	blinkTimeActual	7000			@blinkTin
7						

Index	Time	Node	Message
005	28/07/2021 20:53:27.506	ser1/COM18/MyArduino	Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link
006	28/07/2021 21:00:02.140	ser1/COM18/MyArduino	[Row 6] Illegal variable name '•' specified

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 14404/14404

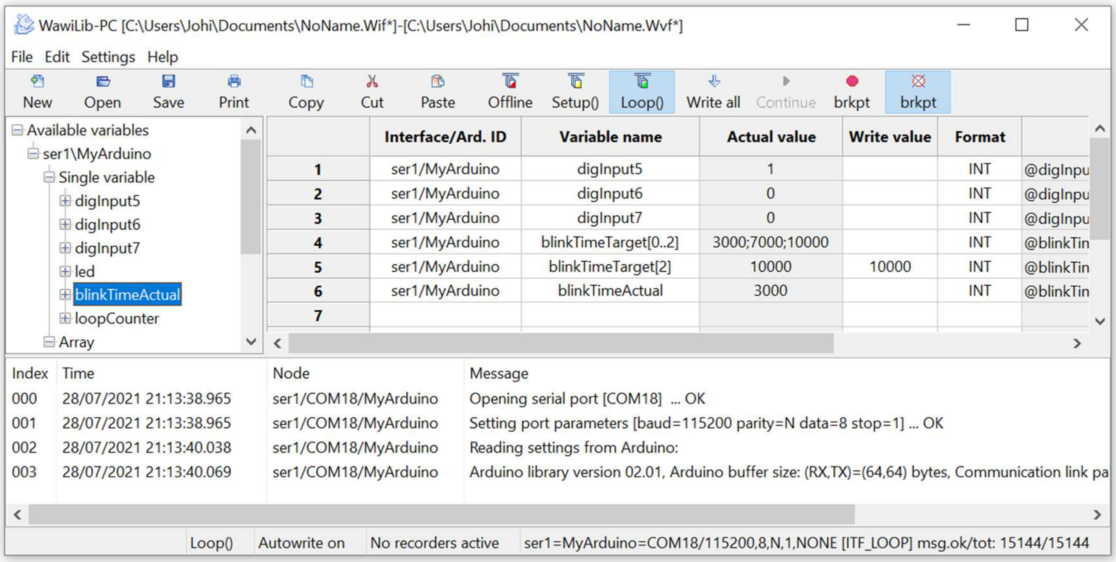
Fig. 3.10. DI 6 at 5V makes blinkTimeActual 7.000 (OK).

⇨ Check the value of blinkTimeActual is and should be 5.000 (= Not OK), bug identified.

```
    if (digInput5)
        blinkTimeActual = blinkTimeTarget[1]; // bug 3: should blinktimeTarget[0]
```

Fig. 3.11. DI 5 at 5V does not make blinkTimeActual 3.000 (Not OK).

- ✓ Press “Offline”
- ✓ Correct the code compile and download.
- ✓ Press “Setup()”.
- ✓ Check the result.



WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	1		INT	@digInpu
2	ser1/MyArduino	digInput6	0		INT	@digInpu
3	ser1/MyArduino	digInput7	0		INT	@digInpu
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTin
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTin
6	ser1/MyArduino	blinkTimeActual	3000		INT	@blinkTin
7						

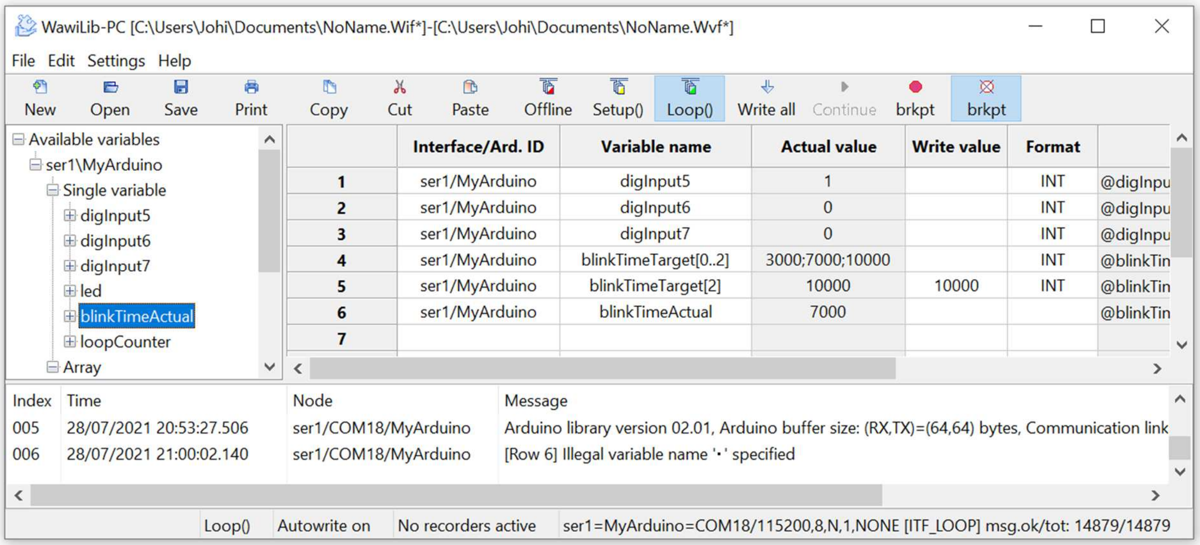
Index	Time	Node	Message
000	28/07/2021 21:13:38.965	ser1/COM18/MyArduino	Opening serial port [COM18] ... OK
001	28/07/2021 21:13:38.965	ser1/COM18/MyArduino	Setting port parameters [baud=115200 parity=N data=8 stop=1] ... OK
002	28/07/2021 21:13:40.038	ser1/COM18/MyArduino	Reading settings from Arduino:
003	28/07/2021 21:13:40.069	ser1/COM18/MyArduino	Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link pa

Loop() Autowrite on No recorders active ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 15144/15144

Fig. 3.12 DI 5 at 5V makes blinkTimeActual 3000 (OK).

3.5. Bug 4: (blinkTimeActual < 0) should be blinkTimeActual > 0

- ✓ Press “Setup()”.


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	1		INT	@digInpu
2	ser1/MyArduino	digInput6	0		INT	@digInpu
3	ser1/MyArduino	digInput7	0		INT	@digInpu
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTin
5	ser1/MyArduino	blinkTimeTarget[2]	10000	10000	INT	@blinkTin
6	ser1/MyArduino	blinkTimeActual	7000			@blinkTin
7						

```
Index  Time                      Node                  Message
005    28/07/2021 20:53:27.506   ser1/COM18/MyArduino  Arduino library version 02.01, Arduino buffer size: (RX,TX)=(64,64) bytes, Communication link
006    28/07/2021 21:00:02.140   ser1/COM18/MyArduino  [Row 6] Illegal variable name '·' specified
```

```
Loop()  Autowrite on  No recorders active  ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 14879/14879
```

Fig. 3.13. WawiBlinkDebugUsb sketch with previous bugs corrected.

- ✓ Connect Digital input 5 to +5V.
- ✓ Connect Digital input 6 to GND.
- ✓ Connect Digital input 7 to GND.
- ✓ Check the value of *blinkTimeActual* (should be decreasing)
- ⇨ *blinkTimeActual* is not decreasing and should be. And LED is not blinking ?
- ✓ Activate the breakpoints by pressing on the button brkpt with the full red dot.

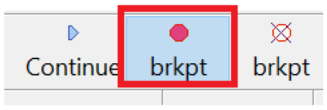
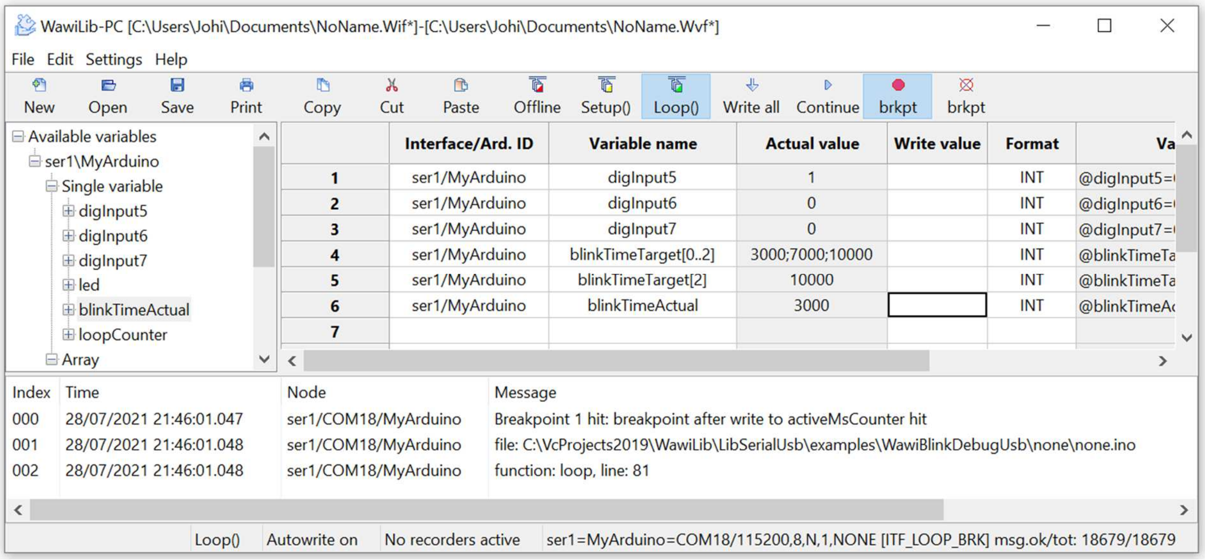


Fig. 3.14. Activate breakpoints


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Va
1	ser1/MyArduino	digInput5	1		INT	@digInput5=
2	ser1/MyArduino	digInput6	0		INT	@digInput6=
3	ser1/MyArduino	digInput7	0		INT	@digInput7=
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTimeTa
5	ser1/MyArduino	blinkTimeTarget[2]	10000		INT	@blinkTimeTa
6	ser1/MyArduino	blinkTimeActual	3000		INT	@blinkTimeAc
7						

```
Index  Time                      Node                  Message
000    28/07/2021 21:46:01.047   ser1/COM18/MyArduino  Breakpoint 1 hit: breakpoint after write to activeMsCounter hit
001    28/07/2021 21:46:01.048   ser1/COM18/MyArduino  file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkDebugUsb\none\none.ino
002    28/07/2021 21:46:01.048   ser1/COM18/MyArduino  function: loop, line: 81
```

```
Loop()  Autowrite on  No recorders active  ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP_BRK] msg.ok/tot: 18679/18679
```

Fig. 3.15.Breakpoint hit at source code line 81.

- ✓ Look at the WawiLib status bar in fig 3.15. status of code execution is ITF_LOOP_BRK (breakpoint hit)
- ✓ Look at line 81: so the execution of the code reaches this point of the source code.

```
    if (digInput5 || digInput6 || digInput7)
    {
81:      WawiSrv.wawiBreak(1, "breakpoint after write to activeMsCounter hit");
    }

    while (blinkTimeActual < 0) // bug 4: should be blinkTimeActual > 0
    {
86:       WawiSrv.wawiBreak(2, "In while loop");
```

Fig. 3.16. Source code with 2 breakpoint locations.

- ✓ And the values of blinkTimeActual at that spot is 3000.
- ✓ Hit the continue button.

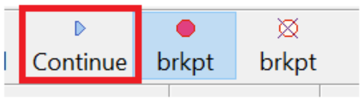
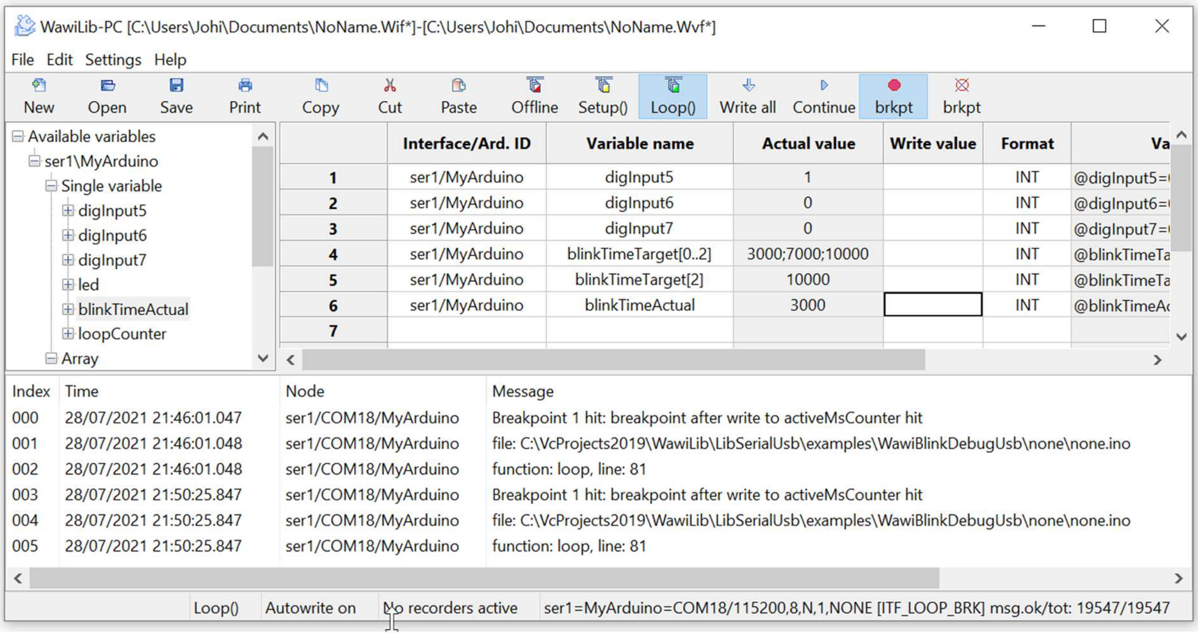


Fig. 3.17. Continue execution after breakpoint.


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	Va
1	ser1/MyArduino	digInput5	1		INT	@digInput5=
2	ser1/MyArduino	digInput6	0		INT	@digInput6=
3	ser1/MyArduino	digInput7	0		INT	@digInput7=
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTimeTa
5	ser1/MyArduino	blinkTimeTarget[2]	10000		INT	@blinkTimeTa
6	ser1/MyArduino	blinkTimeActual	3000		INT	@blinkTimeAc
7						

```
Index  Time                     Node                 Message
000    28/07/2021 21:46:01.047  ser1/COM18/MyArduino Breakpoint 1 hit: breakpoint after write to activeMsCounter hit
001    28/07/2021 21:46:01.048  ser1/COM18/MyArduino file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkDebugUsb\none\none.ino
002    28/07/2021 21:46:01.048  ser1/COM18/MyArduino function: loop, line: 81
003    28/07/2021 21:50:25.847  ser1/COM18/MyArduino Breakpoint 1 hit: breakpoint after write to activeMsCounter hit
004    28/07/2021 21:50:25.847  ser1/COM18/MyArduino file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkDebugUsb\none\none.ino
005    28/07/2021 21:50:25.847  ser1/COM18/MyArduino function: loop, line: 81
```

```
Loop()   Autowrite on   No recorders active   ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP_BRK] msg.ok/tot: 19547/19547
```

Fig. 3.18. Breakpoint at line 81 is hit twice, the one in 86 is not and it should.

- ⇨ A break again at 81 and <u>not</u> at 86, so we are not entering the loop in spite of a correct value for blinkTimeActual at this location.
- ⇨ Bug found, seems < needs to be >.

```
    if (digInput5 || digInput6 || digInput7)
    {
81:      WawiSrv.wawiBreak(1, "breakpoint after write to activeMsCounter hit");
    }

    while (blinkTimeActual > 0) // bug 4: should be blinkTimeActual > 0
    {
86:       WawiSrv.wawiBreak(2, "In while loop");
```

Fig. 3.19. Source code with 2 breakpoint locations.

- ✓ Correct the bug.
- ✓ Press “offline”
- ✓ Compile e Download the corrected code.
- ✓ Press “Setup()”
- ✓ Add the variable *led* to the grid with variables.
- ✓ Connect Digital input 5 to GND.
- ✓ Connect Digital input 6 to GND.
- ✓ Connect Digital input 7 to GND.
- ⇨ Now *blinkTimeActual* is counting down and the LED is blinking.
- ⇨ Enable Display .print() messages in the output window. (right mouse click)

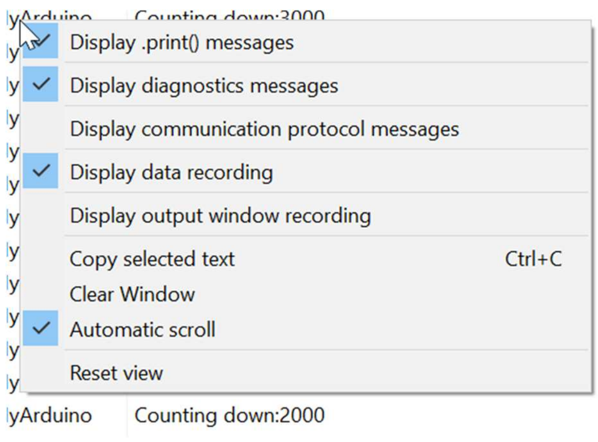
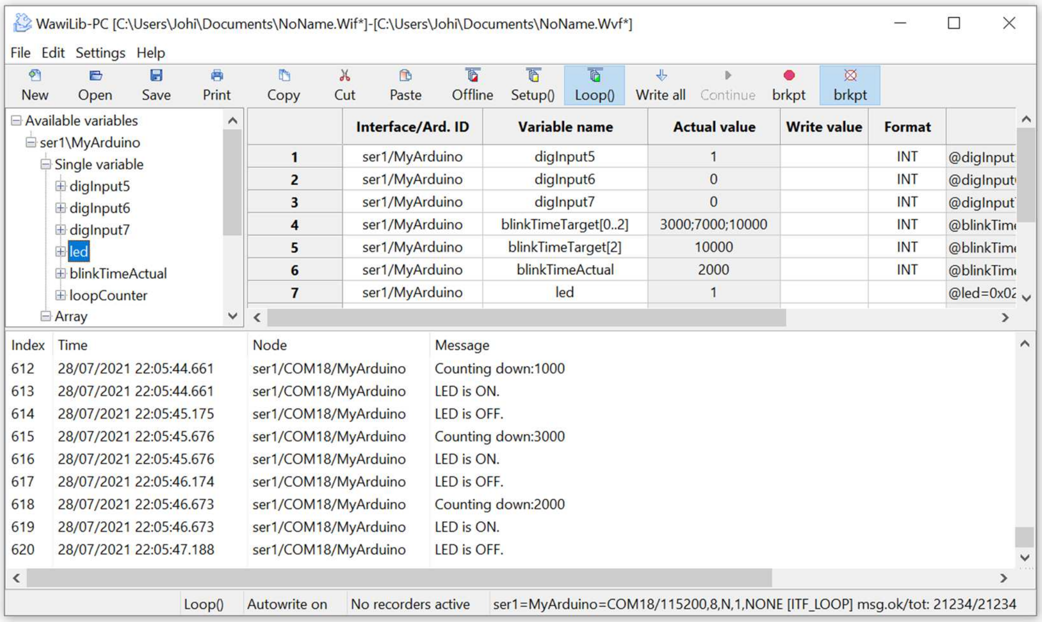

yArduino Counting down:3000
✓ Display .print() messages
✓ Display diagnostics messages
Display communication protocol messages
✓ Display data recording
Display output window recording
Copy selected text Ctrl+C
Clear Window
✓ Automatic scroll
Reset view
yArduino Counting down:2000

Fig 3.20. Enable .print() messages in the output window.

- ⇨ The output window shows the output of the .print() statements
- ⇨ LED blinks on and off
- ⇨ The variable led alternates between 1 and 0.
- ⇨ The status of MyArduino (status bar) is now ITF_LOOP: code in Loop() is executing.


WawiLib-PC [C:\Users\Johi\Documents\NoName.Wif*]-[C:\Users\Johi\Documents\NoName.Wvf*]
File Edit Settings Help
New Open Save Print Copy Cut Paste Offline Setup() Loop() Write all Continue brkpt brkpt
Available variables
ser1\MyArduino
Single variable
digInput5
digInput6
digInput7
led
blinkTimeActual
loopCounter
Array

	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	1		INT	@digInput
2	ser1/MyArduino	digInput6	0		INT	@digInput
3	ser1/MyArduino	digInput7	0		INT	@digInput
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTim
5	ser1/MyArduino	blinkTimeTarget[2]	10000		INT	@blinkTim
6	ser1/MyArduino	blinkTimeActual	2000		INT	@blinkTim
7	ser1/MyArduino	led	1			@led=0x02

```
Index  Time                      Node                   Message
612    28/07/2021 22:05:44.661   ser1/COM18/MyArduino   Counting down:1000
613    28/07/2021 22:05:44.661   ser1/COM18/MyArduino   LED is ON.
614    28/07/2021 22:05:45.175   ser1/COM18/MyArduino   LED is OFF.
615    28/07/2021 22:05:45.676   ser1/COM18/MyArduino   Counting down:3000
616    28/07/2021 22:05:45.676   ser1/COM18/MyArduino   LED is ON.
617    28/07/2021 22:05:46.174   ser1/COM18/MyArduino   LED is OFF.
618    28/07/2021 22:05:46.673   ser1/COM18/MyArduino   Counting down:2000
619    28/07/2021 22:05:46.673   ser1/COM18/MyArduino   LED is ON.
620    28/07/2021 22:05:47.188   ser1/COM18/MyArduino   LED is OFF.
```

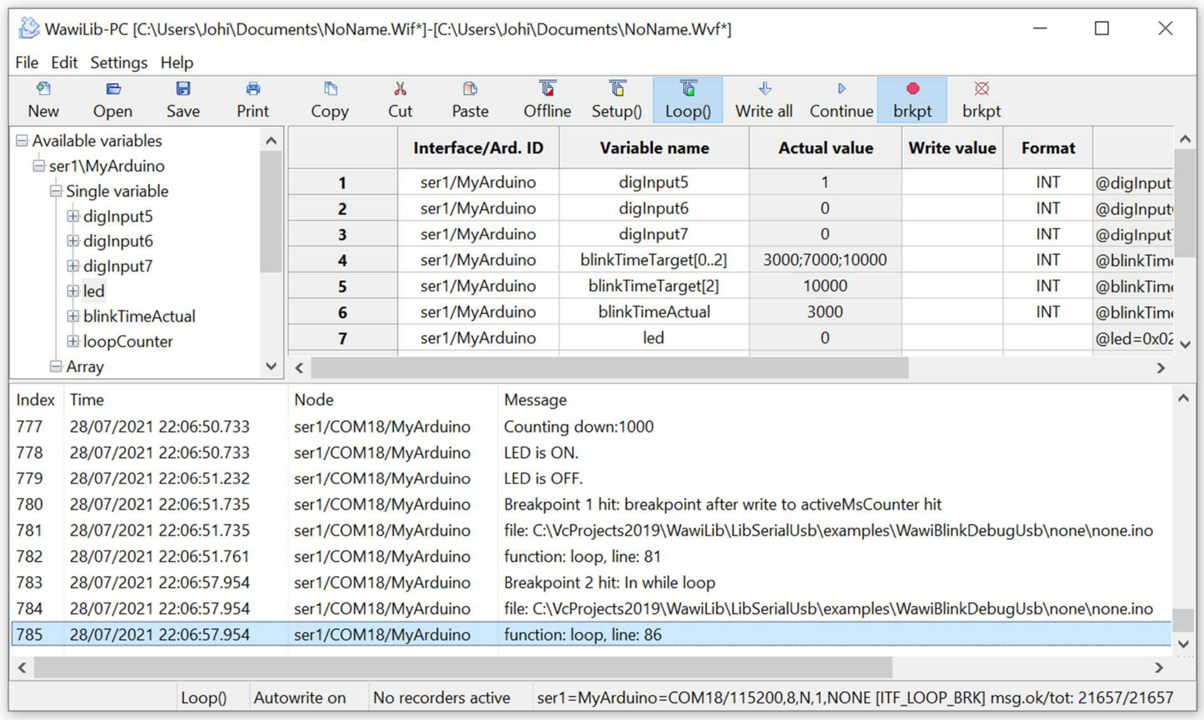
```
Loop()   Autowrite on   No recorders active   ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP] msg.ok/tot: 21234/21234
```

Fig 3.21. Program with code running as it should and debug output in the bottom window.

- ✓ Enable the breakpoints (press “brkpt”).
- ✓ Press “continue” when a breakpoint is hit.

✓ Press “continue” again when a breakpoint is hit.

⇨ In Fig. 3.22. You can see that both breakpoints are used. Bug is indeed corrected.



	Interface/Ard. ID	Variable name	Actual value	Write value	Format	
1	ser1/MyArduino	digInput5	1		INT	@digInput
2	ser1/MyArduino	digInput6	0		INT	@digInput
3	ser1/MyArduino	digInput7	0		INT	@digInput
4	ser1/MyArduino	blinkTimeTarget[0..2]	3000;7000;10000		INT	@blinkTim
5	ser1/MyArduino	blinkTimeTarget[2]	10000		INT	@blinkTim
6	ser1/MyArduino	blinkTimeActual	3000		INT	@blinkTim
7	ser1/MyArduino	led	0			@led=0x02

```
Index Time                     Node                   Message
777   28/07/2021 22:06:50.733  ser1/COM18/MyArduino   Counting down:1000
778   28/07/2021 22:06:50.733  ser1/COM18/MyArduino   LED is ON.
779   28/07/2021 22:06:51.232  ser1/COM18/MyArduino   LED is OFF.
780   28/07/2021 22:06:51.735  ser1/COM18/MyArduino   Breakpoint 1 hit: breakpoint after write to activeMsCounter hit
781   28/07/2021 22:06:51.735  ser1/COM18/MyArduino   file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkDebugUsb\none\none.ino
782   28/07/2021 22:06:51.761  ser1/COM18/MyArduino   function: loop, line: 81
783   28/07/2021 22:06:57.954  ser1/COM18/MyArduino   Breakpoint 2 hit: In while loop
784   28/07/2021 22:06:57.954  ser1/COM18/MyArduino   file: C:\VcProjects2019\WawiLib\LibSerialUsb\examples\WawiBlinkDebugUsb\none\none.ino
785   28/07/2021 22:06:57.954  ser1/COM18/MyArduino   function: loop, line: 86
```

```
Loop()   Autowrite on   No recorders active   ser1=MyArduino=COM18/115200,8,N,1,NONE [ITF_LOOP_BRK] msg.ok/tot: 21657/21657
```

Fig 3.22. Both breakpoints are used after correcting the code.

4. Further Thoughts

This this demonstration of the capabilities of WawiLib shows that it really contains added value when debugging code. The novice and experienced user can use it to improve and correct code. I used it for multiple programs even in a professional environment.

What is also very hand is that WawiLib can be used by the customers to get real time insight in the operation of the code. For example tuning a PID loop of a process that cannot be interrupted (you cannot use the console window to input a new parameter because the process needs to remain controlled) this approach is really adding value to the Arduino experience.

5. Further Reading

This demo demonstrates how to debug an Arduino sketch with WawiLib. The way to go is to visualize all variables of interest on the screen and observe what your application is doing. Debugging goes much faster and is easier because you see real time values of your variables. You can also change values to check if your analysis is correct without recompiling and downloading your program.

Another idea to debug is to record all variables with the WawiLib data recorder and then analyze what happened with Excel or another program of your choice.

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