ABBUC HARDWARE COMPETITION 2014

Marcin Sochacki (Montezuma)

SIO2BT Manual v3.2

Many Thanks to:

Bernd, Bob!k, Dietrich, drac030, FlashJazzCat, Greblus, HardwareDoc, Hias, Igor Gramblička, Kr0tki, Lotharek, mr-atari, Tom Hudson, TRUB, xxl

Table Of Contents

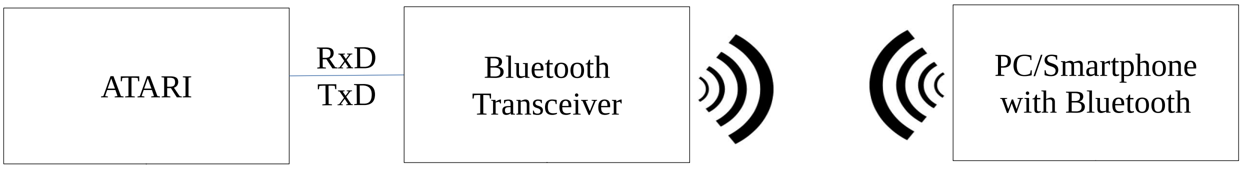
Introduction

SIO2BT project is a set of hardware and software solutions related to the wireless Bluetooth communication between the 8-bit Atari computers and Bluetooth (BT) enabled Serial Input Output (SIO) devices.

Due to delay in the Bluetooth data transmission it was necessary to modify the original Atari OS. For that reason the timeout values (which could not be met) were increased.
A tool for patching Atari OS rom files (for Atari 800 and Atari XL/XE) is provided as a part of the project. Additionally there are disk images available for Atari XL/XE for those who want to use SIO2BT with original, not modified Atari computers.

Another difficulty was missing information about the SIO Command Line status. Hardware handshaking had to be replaced with a software solution for Command Frame detection.

Most of the Bluetooth transceivers available on the market are based on the CSR BC417143 chipset supporting SPP (Bluetooth Serial Port Profile). Two Bluetooth transceivers plugged between any two legacy serial devices would (in theory) be transparent and would appear as if they were connected via cable. In practice missing hardware handshaking and non-deterministic delays cause problems if the protocols rely on timing or on handshaking. And this is exactly the case with the SIO Protocol.



SIO2BT

The BT transceivers provide RxD and TxD lines (TTL voltage level), which can be connected directly to the DATA OUT and DATA IN lines of the Atari SIO port.

Once configured for the SIO Communication (Baudrate = 19200, etc.), a Bluetooth transceiver can be paired with a PC. When a device is paired, the Bluetooth Software Stack on the PC creates a virtual serial port for it. Opening such ports establishes a Bluetooth connection between the PC and the transceiver. A PC can run the software emulating SIO devices.

I have implemented modifications for the open source project RespeQt to support communication via Bluetooth:

- software command frame detection (SOFTWARE handshaking)
- configurable delay for writing data to the Atari

The additional delay for writing is required for Bluetooth, because the Atari OS expects a minimum time delay between the “Acknowledge” Byte and the “Complete” Byte.

This timing is critical and unfortunately can not be guaranteed by Bluetooth. If we send two bytes waiting X milliseconds in between, the first byte can get stuck somewhere in the Bluetooth stack or in the transceiver buffer and the second byte will catch it up so they will arrive one after each other at the Atari. The Atari could skip the second byte and keep waiting for it. Luckily (after a certain timeout) the last Command Frame would be repeated and the communication would continue.

Using a Logic Analyzer and „trial and error“ approach I had to face the fact that there is no 100% guarantee for fast and error free communication. I can recommend a delay value of 10 milliseconds (default setting) for the most cases and bigger values for machines with slower Bluetooth stacks.

Custom SIO code from Dietrich (QMEG OS), HIAS (HI-SPEED OS) and Mr-Atari (MyBios) can handle ACK and COMPETE bytes coming together, so we don't need any delay here.

The SOFTWARE handshaking can also be used (even without any delay) with cheap USB2Serial cables (TTL) that do not support hardware handshaking.

There is one drawback of the SOFTWARE handshaking that should be understood by the users. The emulated SIO device is analyzing data coming from the Atari to detect an incoming Command Frame. We are on the safe side if our device is the only SIO device on the bus. However if the Atari writes data to a different physical SIO devices on the bus, the data (for example a disk sector) can include bytes that build a valid Command Frame. In such cases our device would incorrectly react upon it.

My recommendation is NOT TO USE other physical SIO devices when working with SOFTWARE handshaking, or to use them only to provide data to Atari.

As long as the SIO devices are emulated on a PC, there is no big advantage of wireless communication.

However if SIO devices are emulated on a smartphone, we do not need a PC at all. Many people carry smartphones with them. Games can be downloaded from the internet and immediately loaded to the Atari from an emulated Disk Drive. And I made even one step further and introduced a new SIO Device – an intelligent network device with a TCP/IP stack and a smart device providing time and allowing URL submission (game Hi-Score).

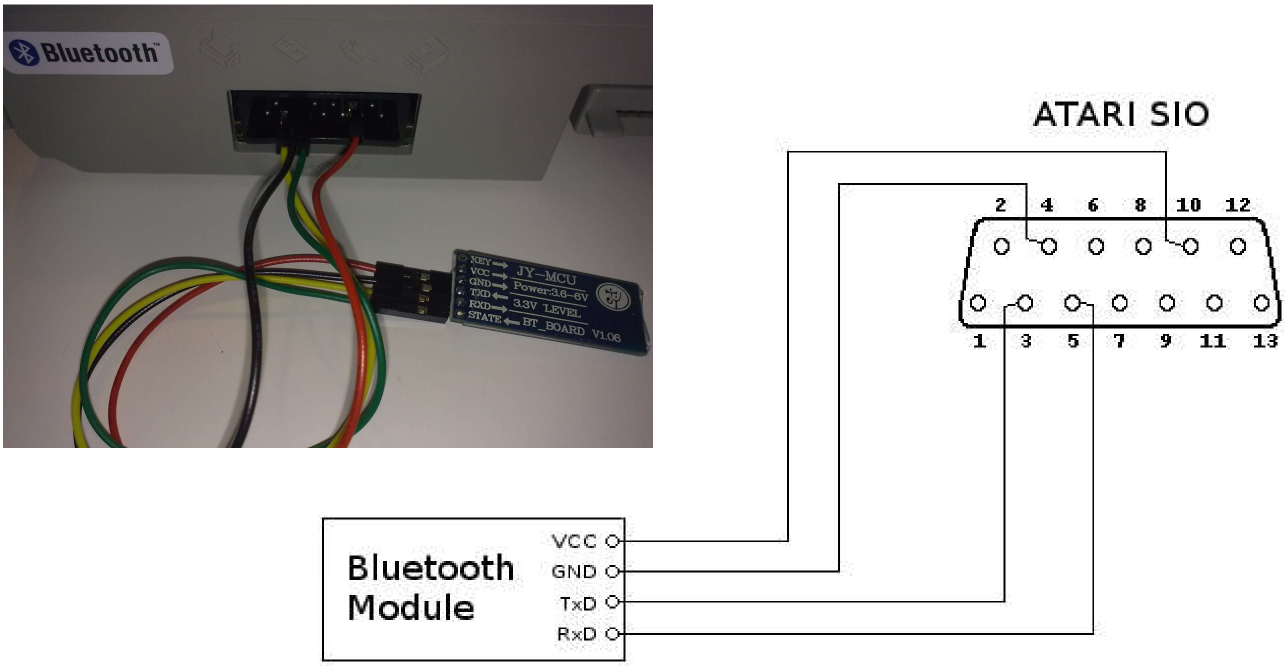
I decided to support the Android platform (the iOS does not support Bluetooth SPP).

The SIO2BT App can be downloaded from the Google Play Store to any smartphone with Android 2.1 (API 7) or higher.

Wiring Diagrams

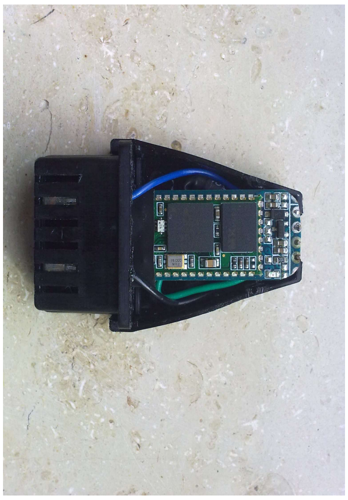
External wiring

If you decide to use SIO2BT like an external Bluetooth dongle, the wiring is pretty straightforward:



(looking at the SIO connector from behind the computer)

The obvious advantage of external wiring is simplicity. The Bluetooth Module is so small that it fits into a SIO Plug.



The hardware is powered from the SIO connector. Every time you switch off the Atari, the Bluetooth Module will be switched off as well and the Bluetooth connection will get lost. It is a disadvantage of this approach, since you have to re-connect after powering on your Atari.

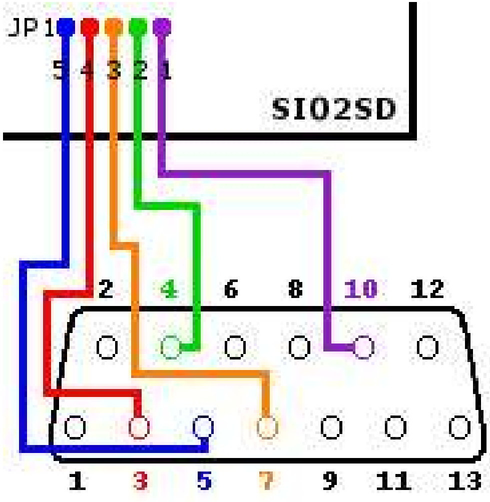
You may think about the Bluetooth connection as a SIO cable. If a SIO cable is not plugged in, an Atari can not communicate with a Disk Drive. The same happens if there is no active Bluetooth connection in case of SIO2BT.

Note: There is intentionally no diode in the Data In line, because the SIO2BT should anyway not be used in a SIO daisy chain with other SIO devices (however it can be optionally added).

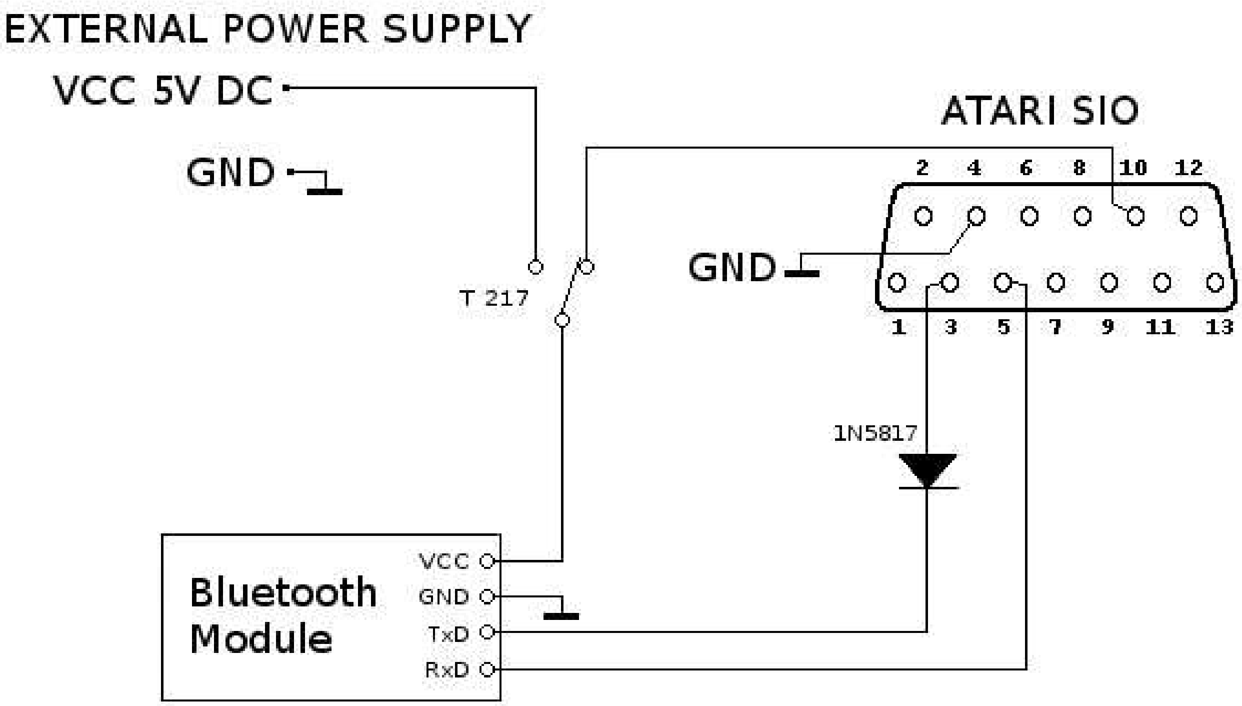
SIO2SD+BT Combo Device Wiring

With basic soldering skills, you will be able to install SIO2BT along with a SIO2SD device.

JP1 SIO2SD	SIO ATARI	BT Module	Description
1	10	VCC	+5V
2	4	GND	GND
3	7	-	COMMAND
4	3	TxD	DATA IN
5	5	RxD	DATA OUT



You may connect the BT module to the SIO PINs or to the SIO2SD JP1 PINs (according to the table above). Remember about a diode between DATA IN and TxD and take care to connect all GND points:



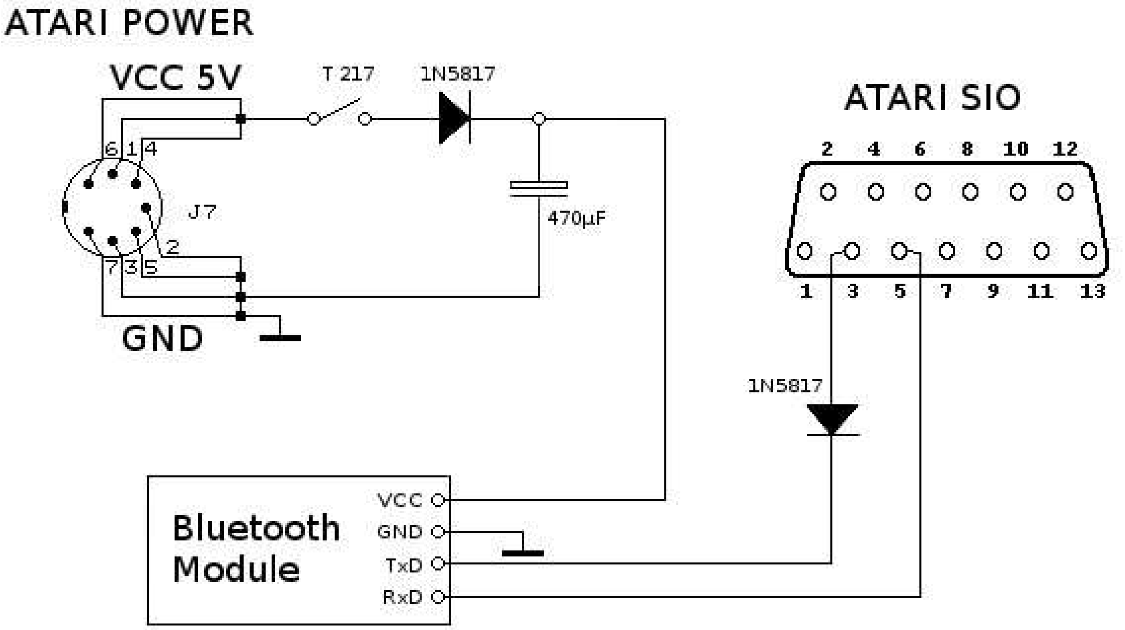
External power supply (5V DC) is optional. It prevents a BT connection drop while powering off the ATARI. If you are going to use the ATARI with the standard OS, the XBIOS loader can be loaded from the SIO2SD. You can attach it to a virtual disk V1 and activate it (SHIFT+K3) for working with SIO2BT. Games in the SIO2BT app should be mounted to D2.

Note 1: Do not assign images to the same disks on SIO2SD and SIO2BT at the same time.

Note 2: Switch off SIO2BT when Atari is writing to other SIO devices.

Internal Wiring

With basic soldering skills, you will be able to install SIO2BT hardware inside your Atari:



The power is grabbed before the main Atari power switch and SIO2BT has its own power switch.

As soon as you connect the power supply unit to the Atari, your Bluetooth Module can be powered on and you can establish a wireless connection.

Switching off the Atari does not drop the Bluetooth connection.

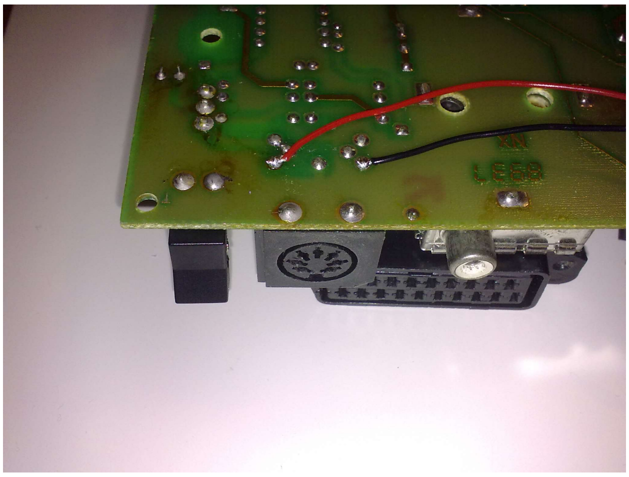
The diode and the capacitor stabilize voltage for the BT Module.

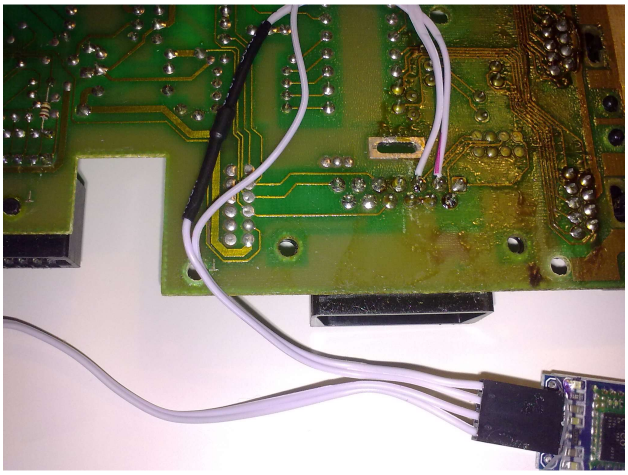
If you do not want to work with SIO2BT you can just switch it off with SIO2BT power switch.

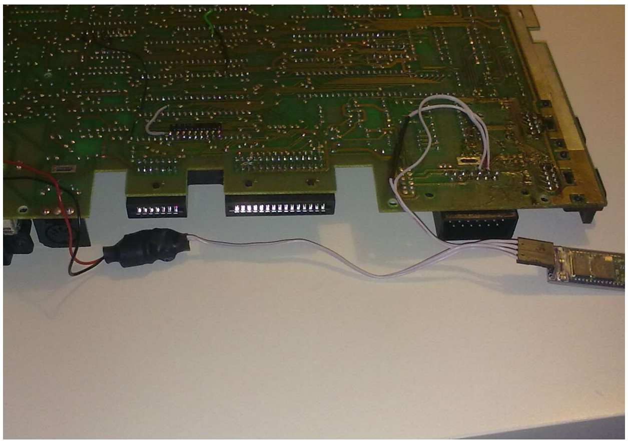
The diode in Data In line makes possible to use SIO2BT with other SIO devices in a daisy chain. This is actually not recommended, however reading data from other SIO devices does not harm SIO2BT.

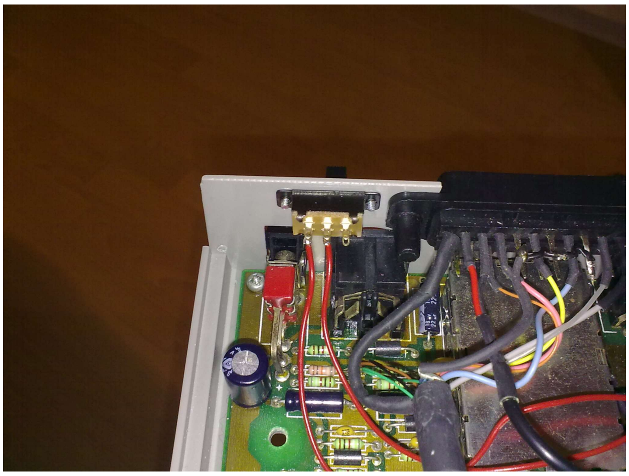
Note1: Switch off SIO2BT when Atari is writing to other SIO devices.

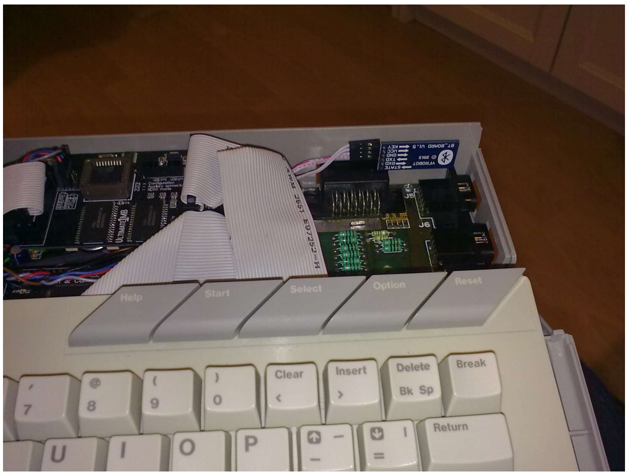
Note2: You can also connect the BT Module's VCC directly to SIO 10 (see external wiring). Since the patched OS allows you to force a COLD START (SHIFT+RESET), you don't need to switch OFF and ON the ATARI (to load a new game) anymore.










Help
Start
Select
Option
Reset
Clear
Insert
Delete
Bk Sp
Break
Return

List of Materials

1) Bluetooth Transceiver

2) Power switch, for example:
T 217 Schiebeschalter-Miniatur, Lötanschluß, 2x UM sw (http://www.reichelt.de/)

3) Capacitor 470 µF, for example:
RAD 470/16 Elektrolytkondensator, 10x12,5mm, RM 5,0mm (http://www.reichelt.de/)

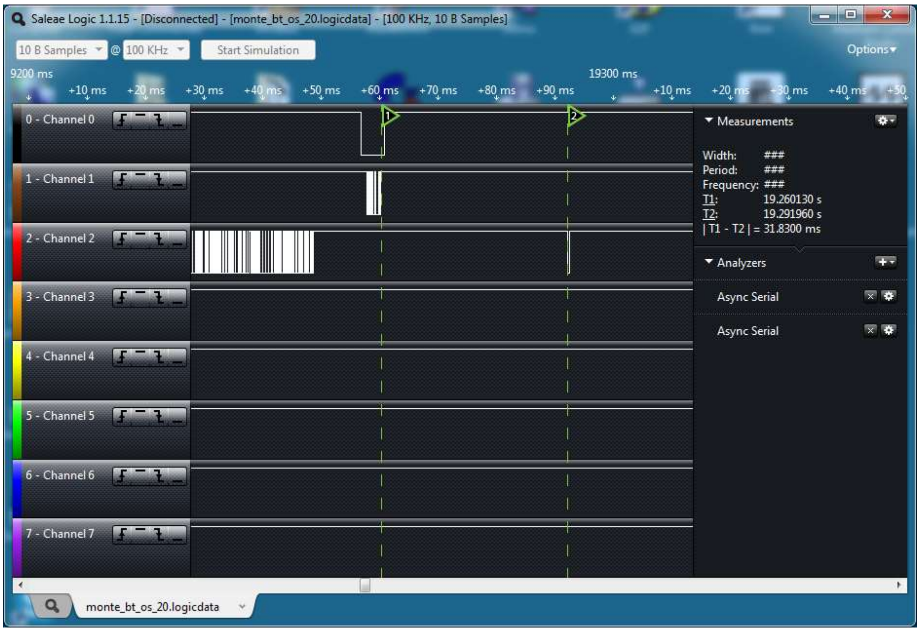
4) 2 x Schottky diode, for example:
1N 5817 Schottky Diode, DO41, 20V, 1A (http://www.reichelt.de/)

5) PC with a Bluetooth dongle (Windows/Linux) or an Android smartphone

Modifications in Atari OS

The SIO Protocol defines the maximum time that Atari should wait for an acknowledge of a command frame. It is 16 milliseconds (for NTSC variant worst case).

Due to delay in the Bluetooth transmission it was necessary to modify these values in the Atari OS.



Saleae Logic 1.1.15 - [Disconnected] - [monte_bt_os_20.logicdata] - [100 KHz, 10 B Samples]

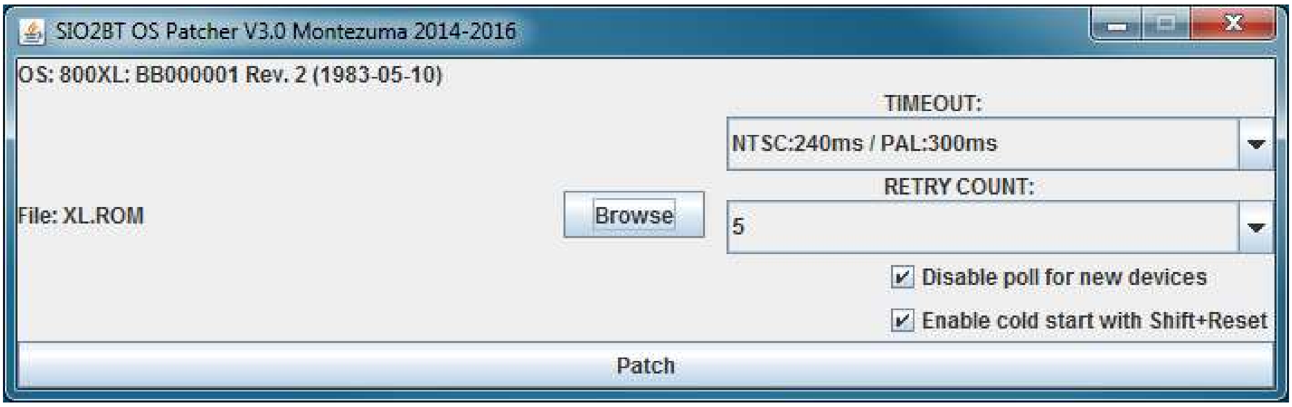
10 B Samples @ 100 KHz

Width: ###
Period: ###
Frequency: ###
T1: 19.260130 s
T2: 19.291960 s
| T1 - T2 | = 31.8300 ms

The biggest delay ever observed with a logic analyzer was about 150 milliseconds, so my recommendation is to increase the timeout above this value. The change influences dramatically the start-up time if no floppy is connected to the Atari. There are two loops in the OS code handling SIO communication. The outer loop handles a complete SIO sequence (command frame / data frame) and is repeated one time per device in error case. The inner loop handles sending a command frame and is repeated up to 13 times in error case (missing Acknowledge Byte). This makes 28 command frames when no disk drive is connected.

There is an extension of the SIO protocol introduced in the XL/XE OS for handling relocatable peripheral handlers. So called “Type 3 Poll Command” is issued at power-on or reset time. This means additional 56 command frames. The only device supporting this new feature is Atari 1090, which exists only as a limited number of prototypes. To preserve the original start-up time and to increase the timeout values at the same time, my recommendation is to to disable the new polling extension and to reduce the number of retries in the inner loop.

With a PC tool ***SIO2BT_OS_Patcher.jar*** you can patch the Atari OS ROM files (for Atari 800 and Atari XL/XE), the Hi-SPEED OS (XL OS with the patch from HIAS) and the QMEG 4.04 ROM file. You just need to download the ROM file and patch it with SIO2BT_OS_Patcher:



OS: 800XL: BB000001 Rev. 2 (1983-05-10)

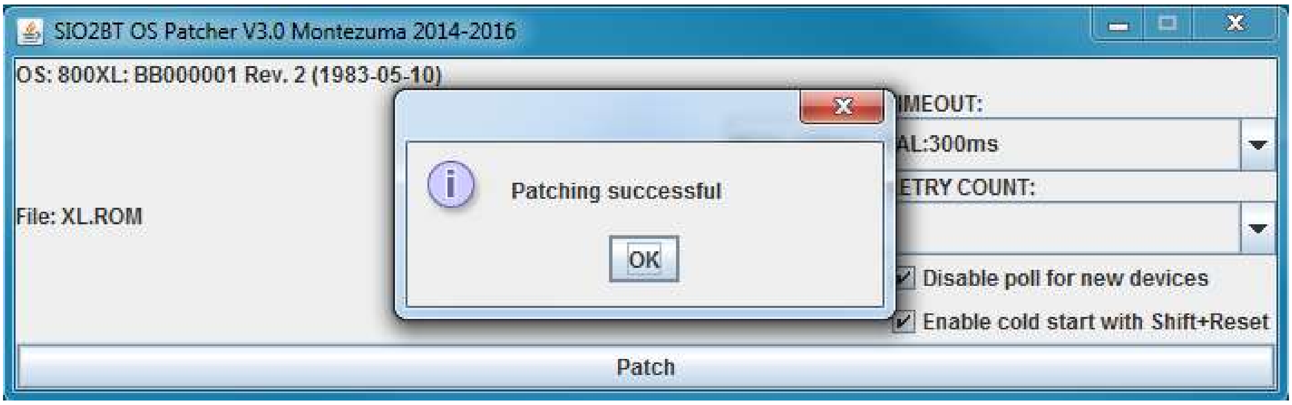
File: XL.ROM

TIMEOUT: NTSC:240ms / PAL:300ms

RETRY COUNT: 5

☑ Disable poll for new devices

☑ Enable cold start with Shift+Reset



OS: 800XL: BB000001 Rev. 2 (1983-05-10)

File: XL.ROM

Patching successful

☑ Disable poll for new devices

☑ Enable cold start with Shift+Reset

The selected ROM file will be cloned and the patched version will be saved under a new name.
Example: XL.ROM > XL.ROM_PATCHED

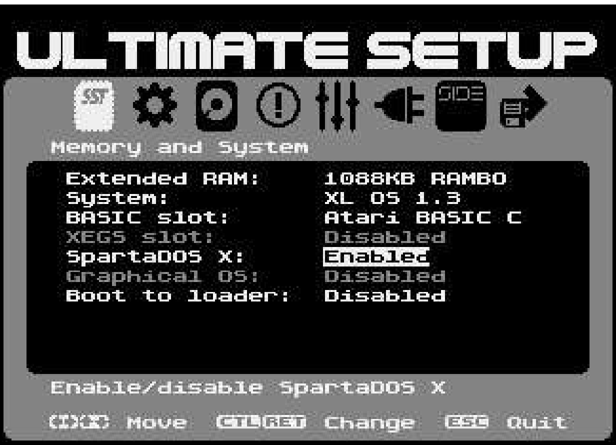
The Atari OS ROMs can be downloaded from:
http://www.ataripreservation.org/websites/freddy.offenga/osromv36.zip (all Atari ROM files)
http://sourceforge.net/projects/atari800/files/ROM/ (Atari XL ROM file)

Bios4Config owners can use Megacart Flash 512K modules to program the flash memory with the updated content (you may use ***Bios4Config.jar*** utility to create a 512K ROM file for flashing).

The owners of very popular ***Ultimate 1MB*** or ***Incognito*** extension boards can upload the patched OS to the board's flash memory with the ***uFlash*** ultility written by FlashJazzCat:
http://www.atari8.co.uk/uflash/index.html

Flashing Ultimate 1MB with BT-enabled OS

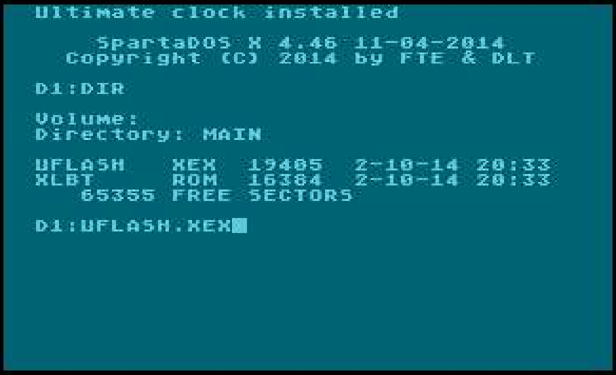
The following screenshots show the procedure step by step.



```
Extended RAM:     1088KB RAMBO
System:           XL OS 1.3
BASIC slot:       Atari BASIC C
XEGS slot:        Disabled
SpartaDOS X:      Enabled
Graphical OS:     Disabled
Boot to loader:   Disabled

Enable/disable SpartaDOS X
```

Go to the Ultimate menu (HELP+RESET) and enable:
- XL OS
- extended memory
- SpartaDOSX.

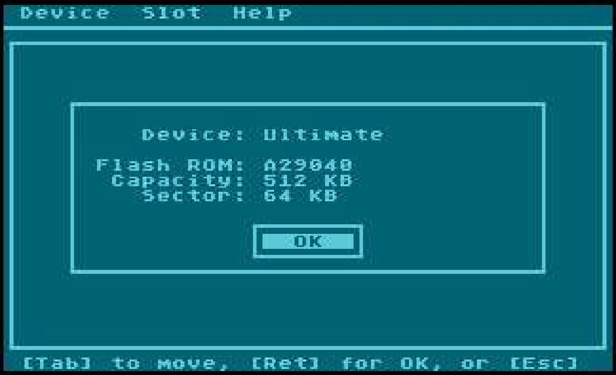
Press S to save settings and Q to Quit



```
Ultimate clock installed

   SpartaDOS X 4.46 11-04-2014
  Copyright (C) 2014 by FTE & DLT

D1:DIR

Volume:
Directory: MAIN

UFLASH    XEX  19405  2-10-14 20:33
XLBT      ROM  16384  2-10-14 20:33
   65355 FREE SECTORS

D1:UFLASH.XEX
```

Prepare an ATR Image containing uflash.xex and the patched OS ROM and mount it as D1.
Using RespeQt you can also mount the directory containing these files as a disk.

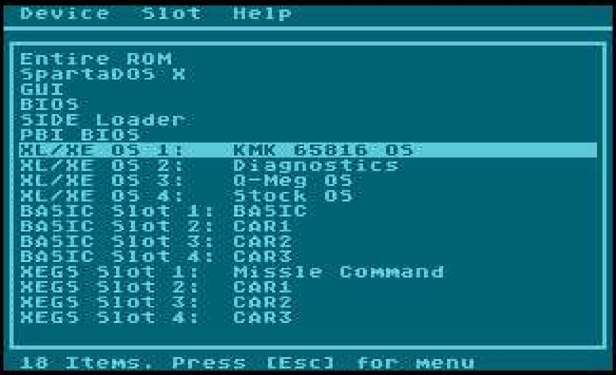
Start uflash.xex



```
    Device: Ultimate

 Flash ROM: A29040
  Capacity: 512 KB
    Sector: 64 KB
```

uflash.xex will automatically detect your board

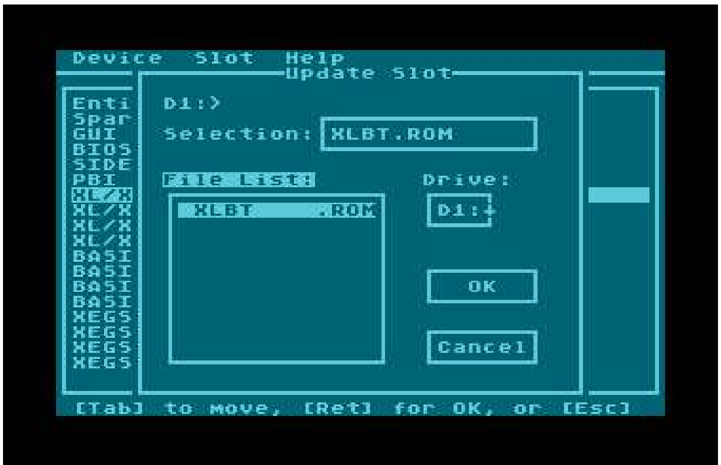
Press „Return“



```
Entire ROM
SpartaDOS X
GUI
BIOS
SIDE Loader
PBI BIOS
XL/XE OS 1:    KMK 65816 OS
XL/XE OS 2:    Diagnostics
XL/XE OS 3:    Q-Meg OS
XL/XE OS 4:    Stock OS
BASIC Slot 1:  BASIC
BASIC Slot 2:  CAR1
BASIC Slot 3:  CAR2
BASIC Slot 4:  CAR3
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3
18 Items. Press [Esc] for menu
```

Select an OS slot, which you want to overwrite.

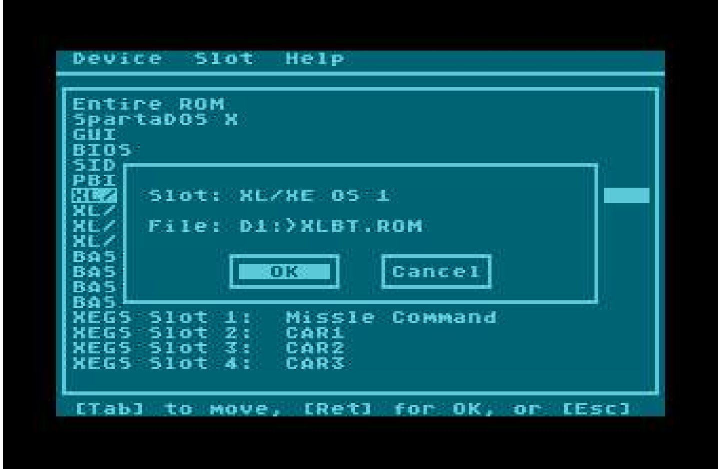
Press „Return“



```
D1:>
Selection: XLBT.ROM
XLBT    .ROM
D1:
[Tab] to move, [Ret] for OK, or [Esc]
```

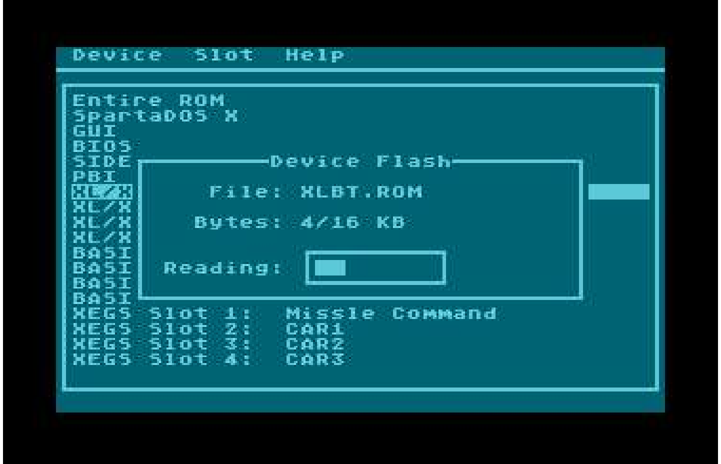
Use arrow keys to select the patched OS ROM file.

Press „Return“



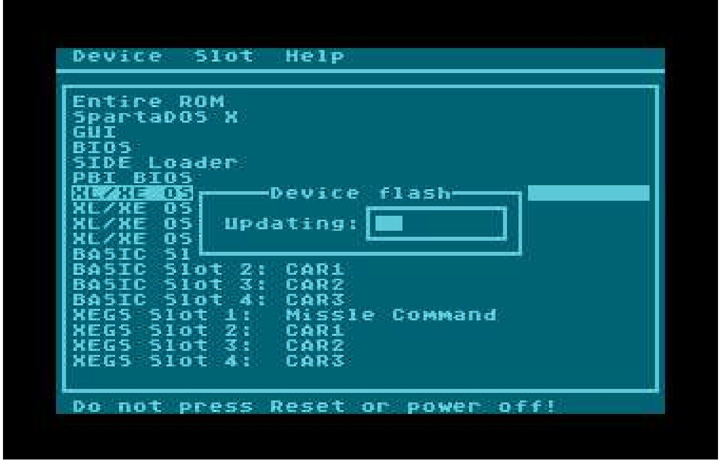
```
Entire ROM
SpartaDOS X
GUI
BIOS
SID
PBI
XL/
XL/
XL/
XL/
BAS
BAS
BAS
BAS
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3
Slot: XL/XE OS 1
File: D1:>XLBT.ROM
[Tab] to move, [Ret] for OK, or [Esc]
```

Press „Return“ to confirm your selection



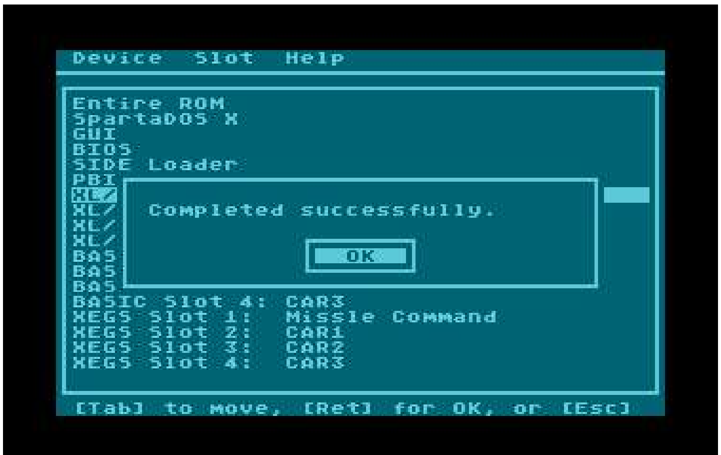
```
File: XLBT.ROM
Bytes: 4/16 KB
Reading:
```

Reading progress is shown.



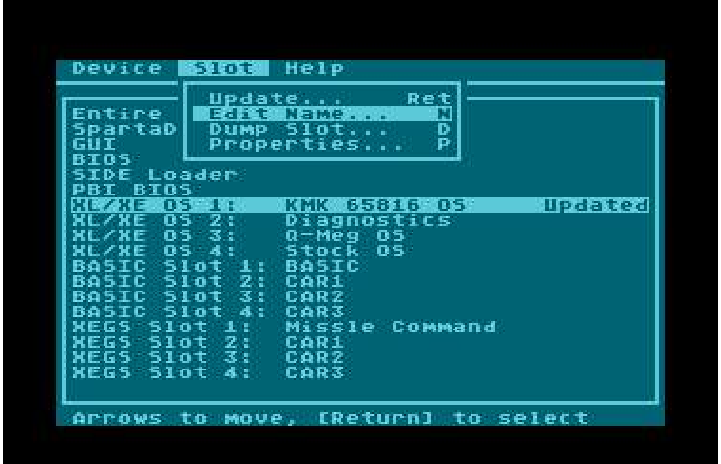
```
Entire ROM
SpartaDOS X
GUI
BIOS
SIDE Loader
PBI BIOS
XL/XE OS
XL/XE OS
XL/XE OS
XL/XE OS
BASIC S1
BASIC Slot 2:  CAR1
BASIC Slot 3:  CAR2
BASIC Slot 4:  CAR3
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3
Updating:
Do not press Reset or power off!
```

Flashing progress is shown.



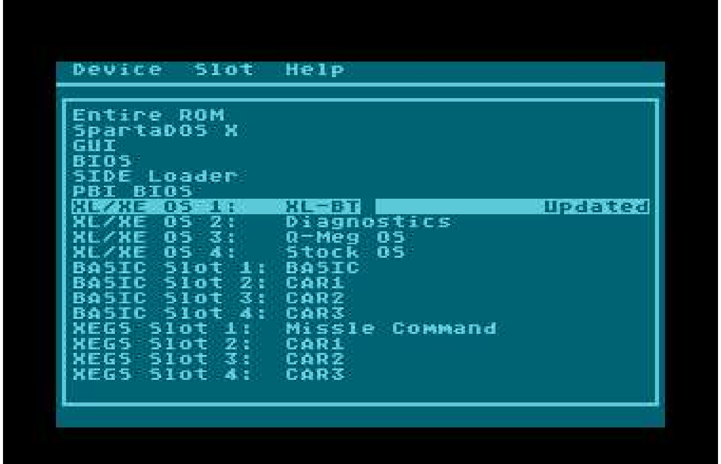
```
Device   Slot   Help

Entire ROM
SpartaDOS X
GUI
BIOS
SIDE Loader
PBI
XL/   Completed successfully.
XL/
XL/
XL/
BAS              OK
BAS
BAS
BASIC Slot 4:  CAR3
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3

[Tab] to move, [Ret] for OK, or [Esc]
```

Press „Return“



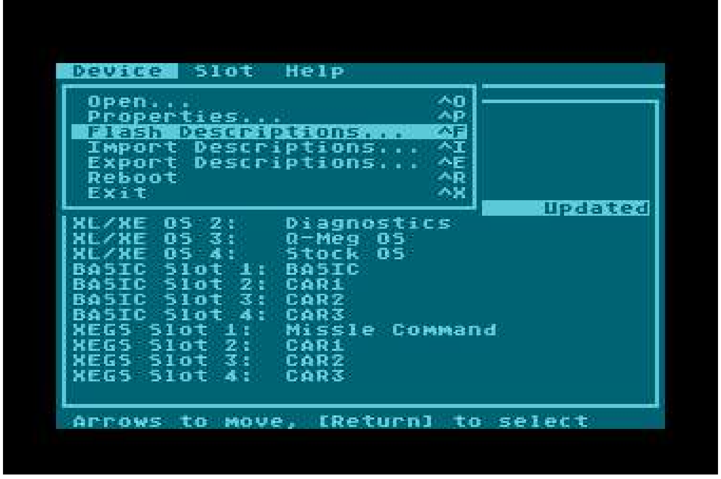
```
Device   Slot   Help
         Update...       Ret
Entire   Edit Name...      N
SpartaD  Dump Slot...      D
GUI      Properties...     P
BIOS
SIDE Loader
PBI BIOS
XL/XE OS 1:    KMK 65816 OS       Updated
XL/XE OS 2:    Diagnostics
XL/XE OS 3:    Q-Meg OS
XL/XE OS 4:    Stock OS
BASIC Slot 1:  BASIC
BASIC Slot 2:  CAR1
BASIC Slot 3:  CAR2
BASIC Slot 4:  CAR3
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3

Arrows to move, [Return] to select
```

Use arrow keys to select the updated OS slot.
Press „Esc“ to show the uflash menu.
Navigate with arrow keys to the„Slot“ menu and select „Edit Name“ item.
Press „Return“



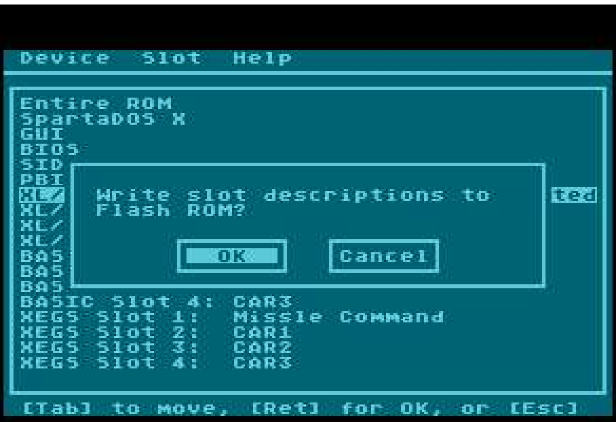
```
Device   Slot   Help

Entire ROM
SpartaDOS X
GUI
BIOS
SIDE Loader
PBI BIOS
XL/XE OS 1:    XL-BT              Updated
XL/XE OS 2:    Diagnostics
XL/XE OS 3:    Q-Meg OS
XL/XE OS 4:    Stock OS
BASIC Slot 1:  BASIC
BASIC Slot 2:  CAR1
BASIC Slot 3:  CAR2
BASIC Slot 4:  CAR3
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3
```

Edit the name and press „Return“



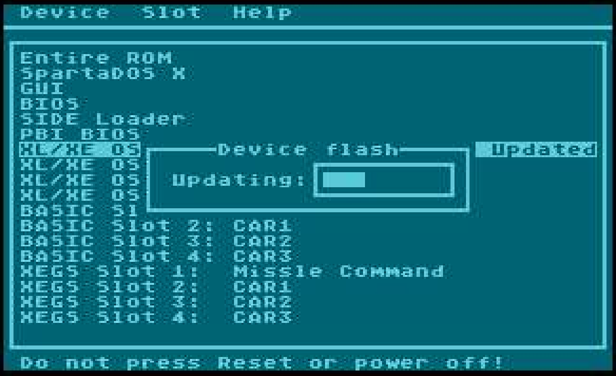
```
Device   Slot   Help
  Open...                  ^O
  Properties...            ^P
  Flash Descriptions...    ^F
  Import Descriptions...   ^I
  Export Descriptions...   ^E
  Reboot                   ^R
  Exit                     ^X
                                  Updated
XL/XE OS 2:    Diagnostics
XL/XE OS 3:    Q-Meg OS
XL/XE OS 4:    Stock OS
BASIC Slot 1:  BASIC
BASIC Slot 2:  CAR1
BASIC Slot 3:  CAR2
BASIC Slot 4:  CAR3
XEGS Slot 1:   Missle Command
XEGS Slot 2:   CAR1
XEGS Slot 3:   CAR2
XEGS Slot 4:   CAR3

Arrows to move, [Return] to select
```

Press „Esc“ to show the uflash menu.
Navigate with arrow keys to the„Device“ menu and select „Flash Descriptions“ item.



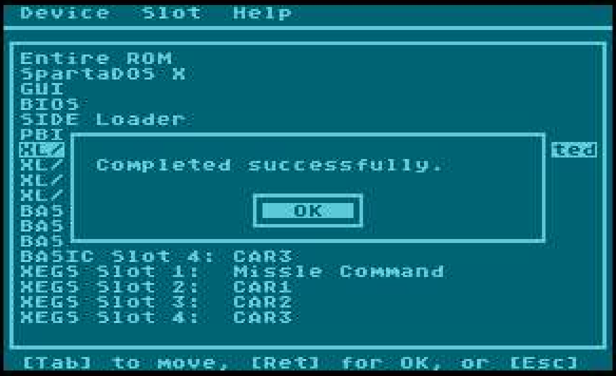
```
Device  Slot  Help

Entire ROM
SpartaDOS X
GUI
BIOS
SID
PBI
XL/   Write slot descriptions to      ted
XL/   Flash ROM?
XL/
XL/
BAS           OK        Cancel
BAS
BAS
BASIC Slot 4: CAR3
XEGS Slot 1:  Missle Command
XEGS Slot 2:  CAR1
XEGS Slot 3:  CAR2
XEGS Slot 4:  CAR3

[Tab] to move, [Ret] for OK, or [Esc]
```

Press „Return“ to confirm



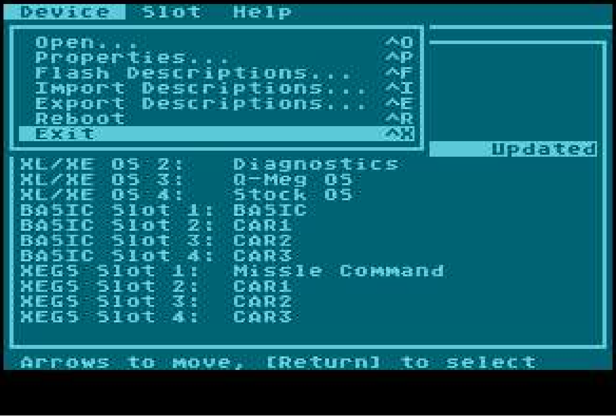
```
Device  Slot  Help

Entire ROM
SpartaDOS X
GUI
BIOS
SIDE Loader
PBI BIOS
XL/XE OS ─Device flash─        Updated
XL/XE OS  Updating:
XL/XE OS
XL/XE OS
BASIC Sl
BASIC Slot 2: CAR1
BASIC Slot 3: CAR2
BASIC Slot 4: CAR3
XEGS Slot 1:  Missle Command
XEGS Slot 2:  CAR1
XEGS Slot 3:  CAR2
XEGS Slot 4:  CAR3

Do not press Reset or power off!
```

Flashing progress is shown.



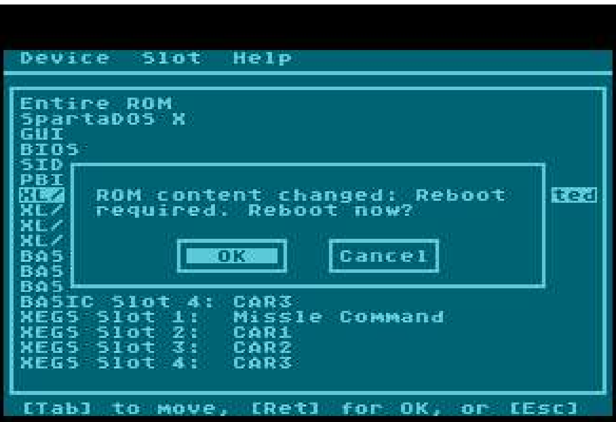
```
Device  Slot  Help

Entire ROM
SpartaDOS X
GUI
BIOS
SIDE Loader
PBI
XL/                                  ted
XL/   Completed successfully.
XL/
XL/
BAS              OK
BAS
BAS
BASIC Slot 4: CAR3
XEGS Slot 1:  Missle Command
XEGS Slot 2:  CAR1
XEGS Slot 3:  CAR2
XEGS Slot 4:  CAR3

[Tab] to move, [Ret] for OK, or [Esc]
```

Press „Return“



```
Device  Slot  Help
  Open...                     ^O
  Properties...               ^P
  Flash Descriptions...       ^F
  Import Descriptions...      ^I
  Export Descriptions...      ^E
  Reboot                      ^R
  Exit                        ^X
                                      Updated
XL/XE OS 2:   Diagnostics
XL/XE OS 3:   Q-Meg OS
XL/XE OS 4:   Stock OS
BASIC Slot 1: BASIC
BASIC Slot 2: CAR1
BASIC Slot 3: CAR2
BASIC Slot 4: CAR3
XEGS Slot 1:  Missle Command
XEGS Slot 2:  CAR1
XEGS Slot 3:  CAR2
XEGS Slot 4:  CAR3

Arrows to move, [Return] to select
```

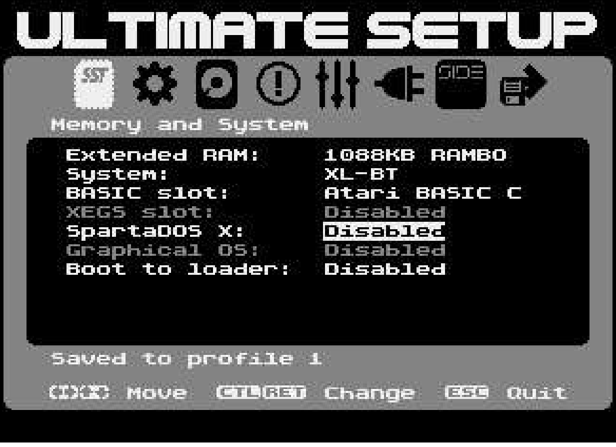
Press „Esc“ to show the uflash menu. Navigate with arrow keys to the„Device“ menu and select „Exit“ item.



```
Device  Slot  Help

Entire ROM
SpartaDOS X
GUI
BIOS
SID
PBI
XL/
XL/
XL/
XL/
BAS
BAS
BAS
BASIC Slot 4: CAR3
XEGS Slot 1:  Missle Command
XEGS Slot 2:  CAR1
XEGS Slot 3:  CAR2
XEGS Slot 4:  CAR3

[Tab] to move, [Ret] for OK, or [Esc]
```

```
ROM content changed: Reboot
required. Reboot now?
```

Press „Return“ to confirm restart.



```
Memory and System

Extended RAM:     1088KB RAMBO
System:           XL-BT
BASIC slot:       Atari BASIC C
XEGS slot:        Disabled
SpartaDOS X:      Disabled
Graphical OS:     Disabled
Boot to loader:   Disabled

Saved to profile 1
```

Go to the Ultimate menu (HELP+RESET) and:
- disable SpartaDOSX.
- select XL-BT OS

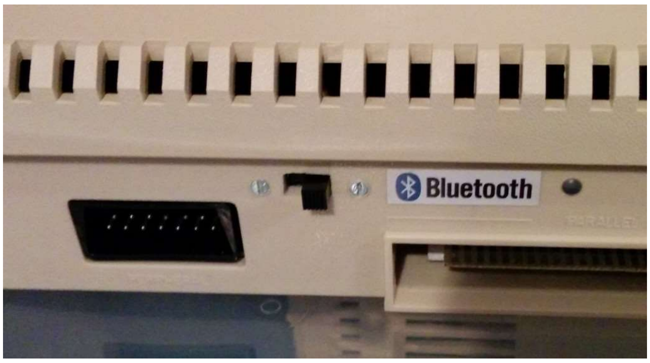
Press S to save settings and C to quit.
Your Atari is ready for SIO2BT :)

EPROM based OS Switch

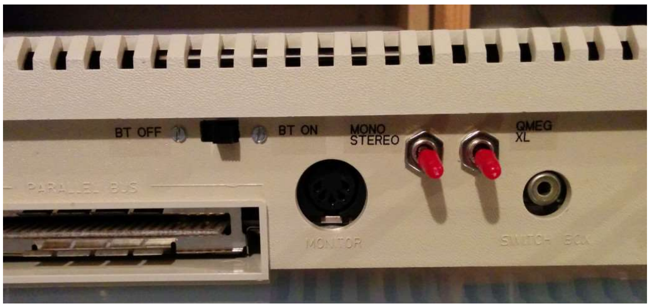
The original OS ROM can be replaced with the following 28-dip EPROMs:

1. 27C128 (size: 16kB = 1 x OS). It is a simple replacement of the original ROM. If you decide for this solution, I recommend to use the XL OS patched for Bluetooth, which offers the highest compatibility with existing software. You do not need any mechanical switch since there will be only one OS in the computer.

2. 27C256 (size: 32kB = 2 x OS). It can host two Operating Systems. You can go for two different OSs patched for Bluetooth (for example XL_BT and QMEG_BT) or you can decide to use the original XL OS and the second OS, patched for Bluetooth (XL_BT or QMEG_BT). If you decide for the second approach, you can re-use the switch T217 (the first row for powering the BT module, the second row for switching the OS).



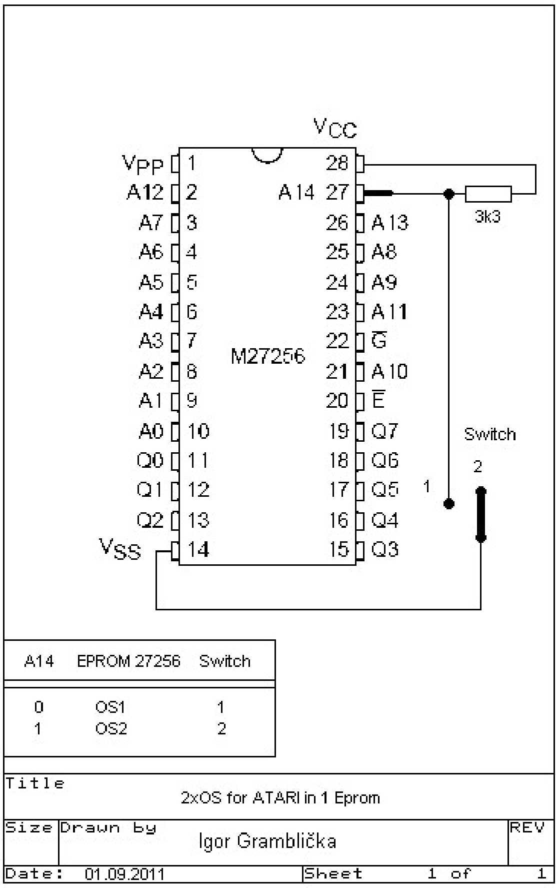
3. 27C512 (size: 64kB = 4 x OS). It can host four Operating Systems. Of course it can be programmed with four dfifferent OSs, but this EPROM gives you a nice possibility to have two sets of operating systems: one set with the original ROMs and one set with the patched ROMs. A mechanical OS switch is used to switch between two OS sets (for example XL and QMEG) and the re-used T217 switches between the original and the patched version (of the OS selected with the first switch) at the time you power the Bluetooth module on and off.



The binary file for programming the EPROM can be created with the following commands:

1. under Windows
   ```
   copy /B QMEG.ROM+QMEGBT.ROM+XL.ROM+XLBT.ROM OUT.ROM
   ```
2. under Linux
   ```
   cat QMEG.ROM QMEGBT.ROM XL.ROM XLBT.ROM > OUT.ROM
   ```

The electrical diagrams for 27C256 and 27C512 (thanks to Igor Gramblička):

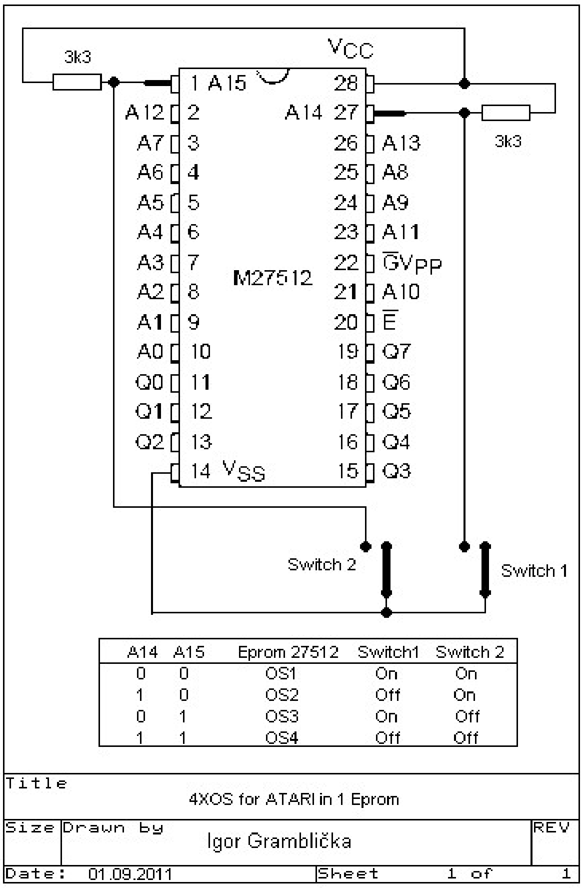


A14	EPROM 27256	Switch
0	OS1	1
1	OS2	2

Title 2xOS for ATARI in 1 Eprom

Size | Drawn by Igor Gramblička | REV

Date: 01.09.2011 | Sheet 1 of 1



A14	A15	Eprom 27512	Switch1	Switch 2
0	0	OS1	On	On
1	0	OS2	Off	On
0	1	OS3	On	Off
1	1	OS4	Off	Off

Title 4XOS for ATARI in 1 Eprom

Size | Drawn by Igor Gramblička | REV

Date: 01.09.2011 | Sheet 1 of 1

Forcing Atari COLDSTART with a patched OS

QMEG OS users may toggle with the TAB key between WARM and COLD reset settings. Depending on this setting the ATARI will perform a warm or a cold reset, when the RESET key is pressed.

If you use the **ATARI OS** ROMs patched with the *SIO2BT_OS_Patcher.jar* tool, you can force a COLDSTART with the **SHIFT+RESET** key combination.

This possibility is specially useful for those who use SIO2BT in a SIO plug. You do not need to power off the ATARI anymore before loading a new game. The Bluetooth connection remains active while the cold start does the job.

Note: With the Ultimate's (Incognito's) BIOS you can hit 'C' (while the board's menu is displayed) to force on OS cold restart.

Ultimate 1MB support for SIO2BT

The „September 2016 Update“ for the Ultimate 1MB provides excellent support for SIO2BT.
A prerequisite for that is an OS supporting PBI devices (for example the XL/XE OS).

When „PBI BIOS“ is enabled and the SIO driver is set to „HSIO“ or „HSIO+SIO2BT“, all IO requests are taken over by the PBI SIO driver (they won't be passed to the OS SIO code at all).

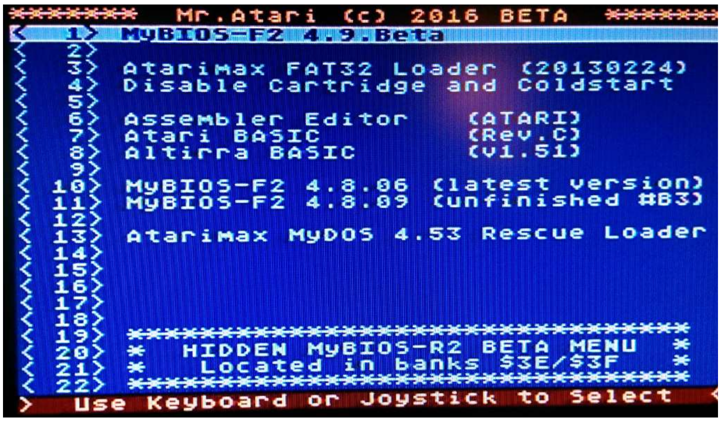
Very interesting is the HSIO+SIO2BT mode, which supports (besides Hi-Speed devices) SIO2BT with the following baudrates: 57600, 38400, 19200.

Please refer to the Ultimate 1MB documentation from FJC for more details.

My IDE II support for SIO2BT

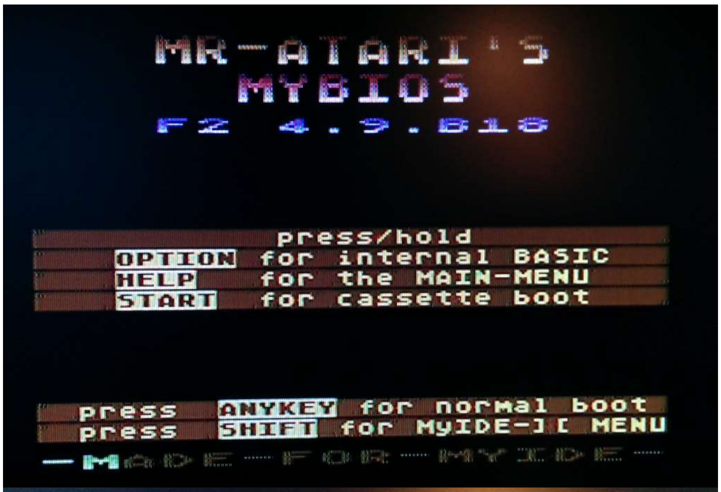
Since „4.9 BETA#17“ Firmware, MyIDE II provides support for SIO2BT.
„Mr-Atari“ (developer of MyIDE II) has re-worked the SIO code to support Bluetooth communication with the following baud rates: 57600, 38400, 19200.

Most of the MyIDE II users just load games from the CF card with the FAT32 Loader, but there is much more to discover. For example you can use MyBIOS to load games over SIO2BT.



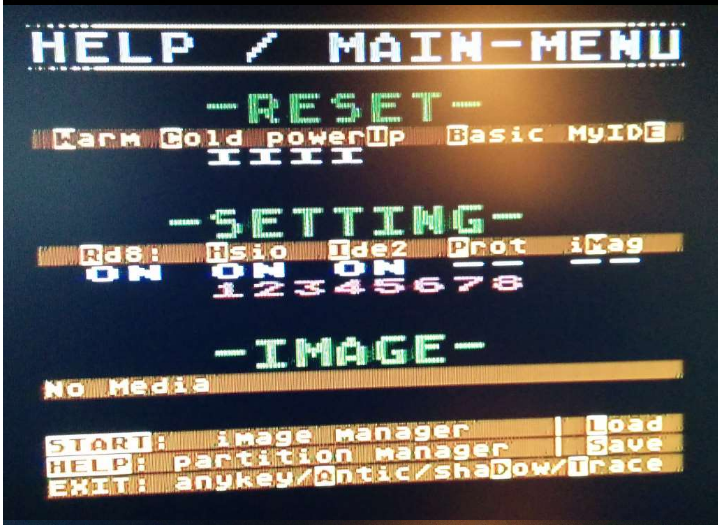
```
*******   Mr.Atari (c) 2016 BETA   *******
<  1> MyBIOS-F2 4.9.Beta
<  2>
<  3> Atarimax FAT32 Loader (20130224)
<  4> Disable Cartridge and Coldstart
<  5>
<  6> Assembler Editor    (ATARI)
<  7> Atari BASIC         (Rev.C)
<  8> Altirra BASIC       (v1.51)
<  9>
< 10> MyBIOS-F2 4.8.06 (latest version)
< 11> MyBIOS-F2 4.8.09 (unfinished #B3)
< 12>
< 13> Atarimax MyDOS 4.53 Rescue Loader
< 14>
< 15>
< 16>
< 17>
< 18>
< 19> ******************************
< 20> *  HIDDEN MyBIOS-R2 BETA MENU  *
< 21> *   Located in banks $3E/$3F  *
< 22> ******************************
>  Use Keyboard or Joystick to Select
```

Select item: „MyBIOS-F2 4.9 Beta“



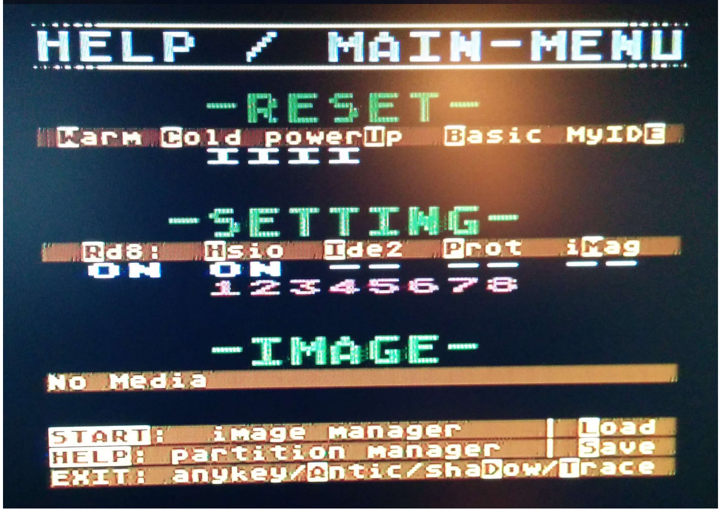
```
MR-ATARI'S
MYBIOS
F2 4.9.B18

press/hold
OPTION for internal BASIC
HELP   for the MAIN-MENU
START  for cassette boot

press  ANYKEY for normal boot
press  SHIFT for MyIDE-][ MENU
-MADE-FOR-MYIDE-
```

Press and hold „HELP“ key



```
HELP / MAIN-MENU
-RESET-
Warm Cold powerUp  Basic MyIDE
      IIII
-SETTING-
Rd8:  Hsio  Ide2  Prot  iMag
 ON    ON    ON    --    --
      12345678
-IMAGE-
No Media
START: image manager      | Load
HELP: partition manager   | Save
EXIT: anykey/Antic/shaDow/Trace
```

Press 'I' to disable IDE2 Harddrive



```
HELP / MAIN-MENU
-RESET-
Warm Cold powerUp  Basic MyIDE
      IIII
-SETTING-
Rd8:  Hsio  Ide2  Prot  iMag
 ON    ON    --    --    --
      12345678
-IMAGE-
No Media
START: image manager      | Load
HELP: partition manager   | Save
EXIT: anykey/Antic/shaDow/Trace
```

Press 'C' to execute a cold start.
The ATARI will boot now from D1 using Bluetooth compatible MyBIOS SIO procedures (you do not need to press OPTION to disable BASIC – BASIC status can be changed by pressing 'B' before triggering a cold start).

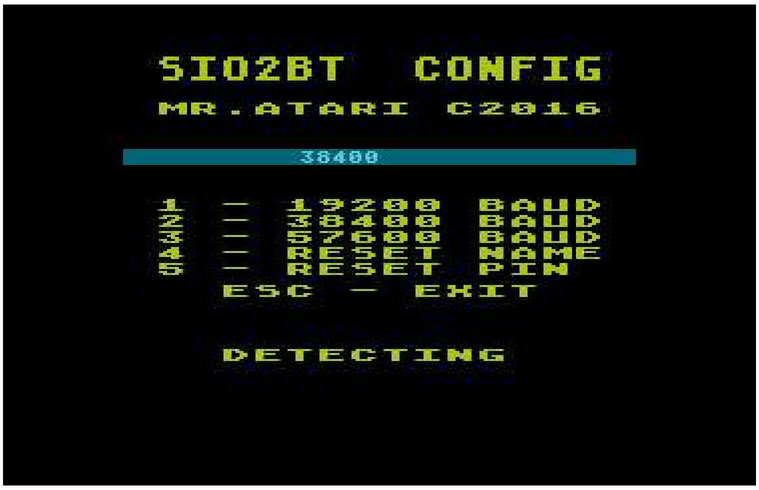
Bluetooth Module configuration

When the Bluetooth connection is not active (LED is blinking) it is possible to change configuration of the module.

There are two ATARI programs serving this purpose:

1) *BTCONFIG* utility, developed by Mr-Atari.

You can change the baudrate, Bluetooth friendly name and the PIN code.



```
SIO2BT  CONFIG
MR.ATARI  C2016

     38400

1 - 19200  BAUD
2 - 38400  BAUD
3 - 57600  BAUD
4 - RESET  NAME
5 - RESET  PIN
  ESC  -  EXIT

  DETECTING
```

2) *btcfg* SpartaDOSX command line utility, developed by FJC

Start with no arguments for a list of switches.

Note: All settings are persisted and SIO2BT supports only one baudrate at a time.
Once configured for 57600 baud, it won't support communication with 19200 baud until it is configured for that baudrate (next time).

Patching Atari OS at runtime

Additionally there are disk images available for Atari XL/XE for those who want to use (with some limitations) SIO2BT with original, not modified Atari computers.
You can use the following images: 600XL.atr, 1200XL(a).atr, 1200XL(b).atr, XL_XE_XEGS.atr.
Mount one of them (depending on your Atari) as D1 and a game to be loaded as D2.
Additionally there is one more set of disk images, which can be used to load ATR games:
600XL_SWAP.atr, 1200XL(a)_SWAP.atr, 1200XL(b)_SWAP.atr, XL_XE_XEGS_SWAP.atr.
Mount one of them (depending on your Atari) as D1 and a game to be loaded as D2.
When the loader is executed, the screen color changes to dark blue. Now SWAP the disks on the Smartphone, so the game to be loaded is mounted as D1 and press any key on the ATARI to continue.
These small (144 bytes) bootable disk images contain only one sector with code for OS patching.
Atari sends two SIO commands to an emulated floppy (D1) at start-up: $53 Get Status and $52 Read Sector 1. The most of the times the communication is successful (with standard SIO) and only sometimes Atari gets stuck for a while (and re-tries). When the requested sector is loaded, the OS SIO code is modified and the Atari continues to load from the second disk (D2).

However the best way to use SIO2BT with the original OS is the **XBIOS loader for bluetooth**.

The loader XBIOS4BT_700.atr (or XBIOS4BT_400.atr) can be mounted as D1 and the game(s) to be loaded as D2. The only difference between the two XBIOS versions is the address in memory, where the loader is located ($700 or $400).

Note1: Some games can not be loaded this way
Note2: The highest compatibility can be achieved with a patched OS (recommended way)

Introduction to Bluetooth

The scope of the Bluetooth specifications is very wide. It covers hardware and software starting from the radio signal to the high level protocols for wireless, short-range communication.

Every Bluetooth enabled device has so called Bluetooth stack (realized partly in the hardware and partly in the software), which implements the Bluetooth specifications (usually only a subset of them). One of the most popular Bluetooth protocols is RFCOMM (Radio Frequency Communications). It was designed to emulate RS-232 serial ports, the SPP (Serial Port Profile) is based on it and it is available in virtually every Bluetooth stack (in particular in a Bluetooth Transceiver used in the SIO2BT project).

In order to establish a wireless connection between two Bluetooth devices, one of the devices has to enter discoverable mode and the other device has to perform a search (device discovery).

In our case the Bluetooth Transceiver automatically enters discoverable mode when powered on and we have to search for it.

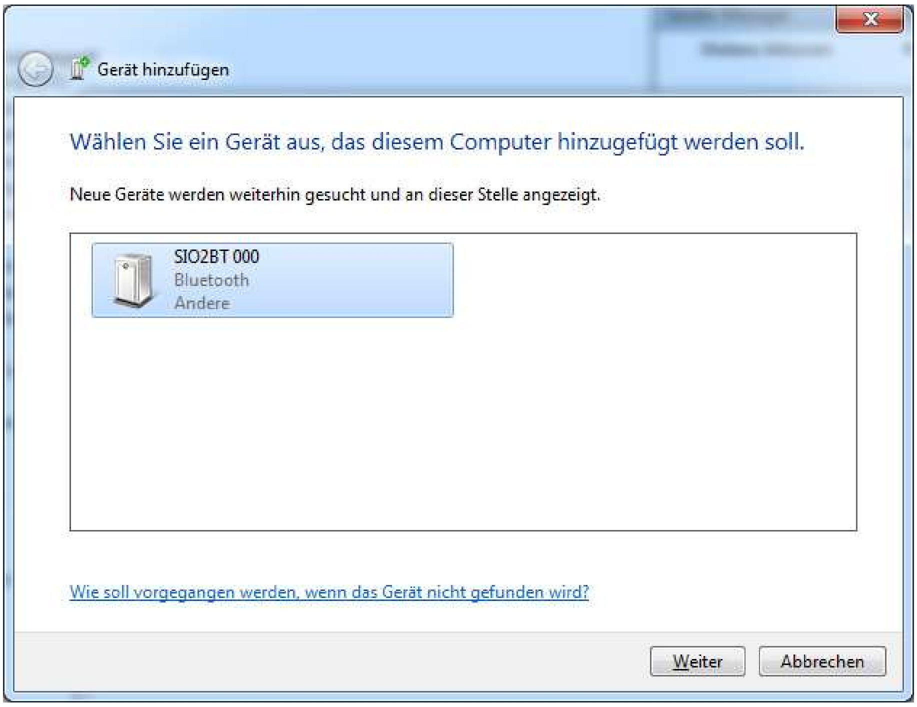
The first step in establishing the connection is an authentication. If devices do not know each other, they have to be paired first.

Pairing requires entering 4-digit PIN code of the Bluetooth Transceiver. After this number is entered both devices generate and store so called “Link Key” for future authentication.

Once devices are paired, the wireless connection can be established. Under Windows and under Linux, the Bluetooth connection is established when a virtual serial port (created after pairing) is opened.

Bluetooth under Windows

Pairing process under Windows 7 is presented on the screen shots below:



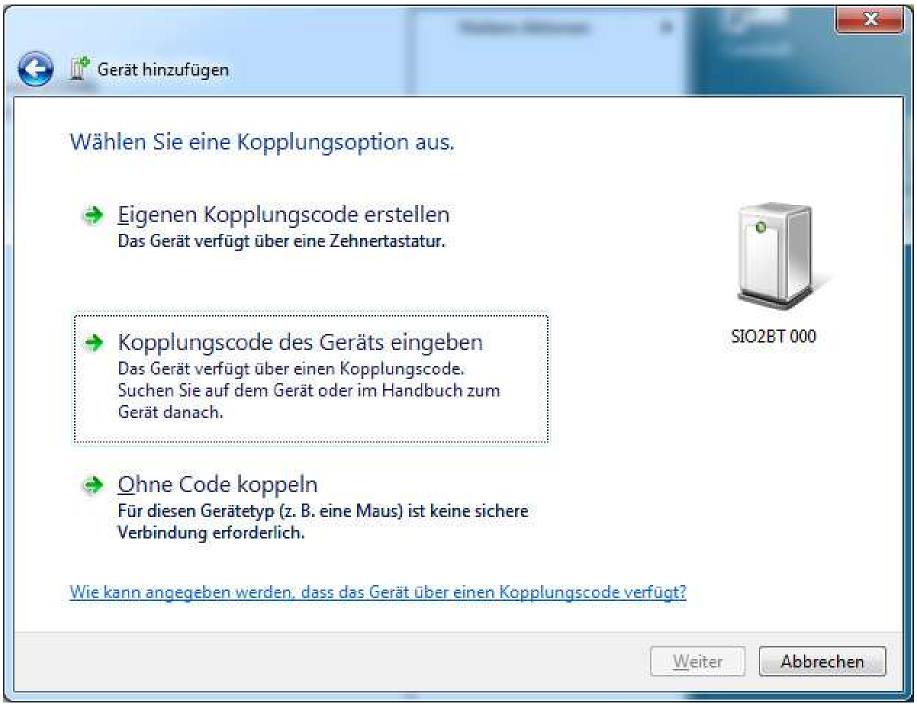
Gerät hinzufügen

Wählen Sie ein Gerät aus, das diesem Computer hinzugefügt werden soll.

Neue Geräte werden weiterhin gesucht und an dieser Stelle angezeigt.

SIO2BT 000
Bluetooth
Andere

Wie soll vorgegangen werden, wenn das Gerät nicht gefunden wird?



Gerät hinzufügen

Wählen Sie eine Kopplungsoption aus.

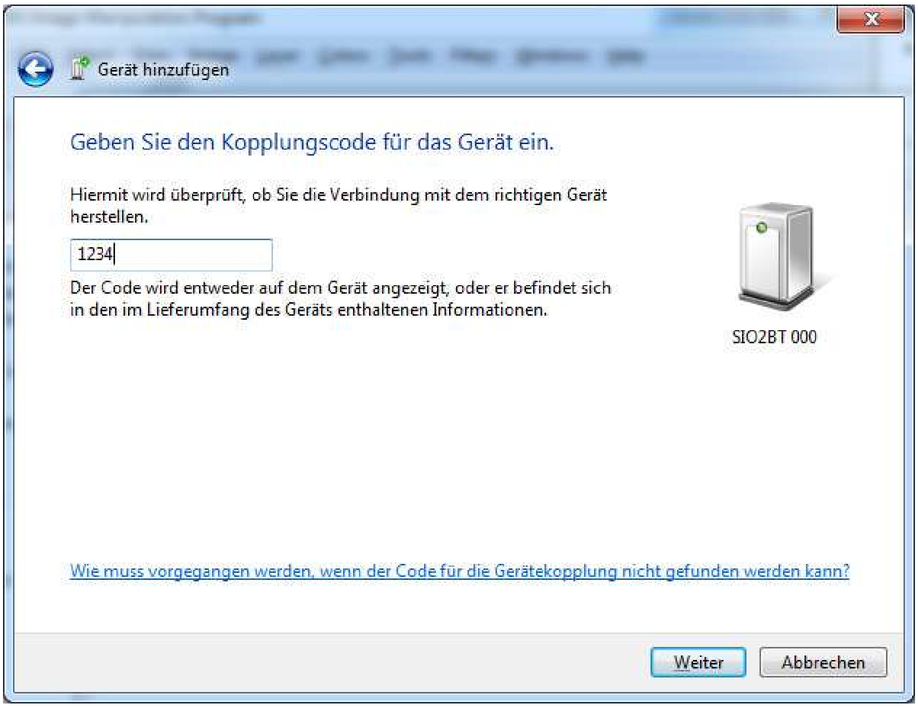
Eigenen Kopplungscode erstellen
Das Gerät verfügt über eine Zehnertastatur.

Kopplungscode des Geräts eingeben
Das Gerät verfügt über einen Kopplungscode. Suchen Sie auf dem Gerät oder im Handbuch zum Gerät danach.

Ohne Code koppeln
Für diesen Gerätetyp (z. B. eine Maus) ist keine sichere Verbindung erforderlich.

SIO2BT 000

Wie kann angegeben werden, dass das Gerät über einen Kopplungscode verfügt?



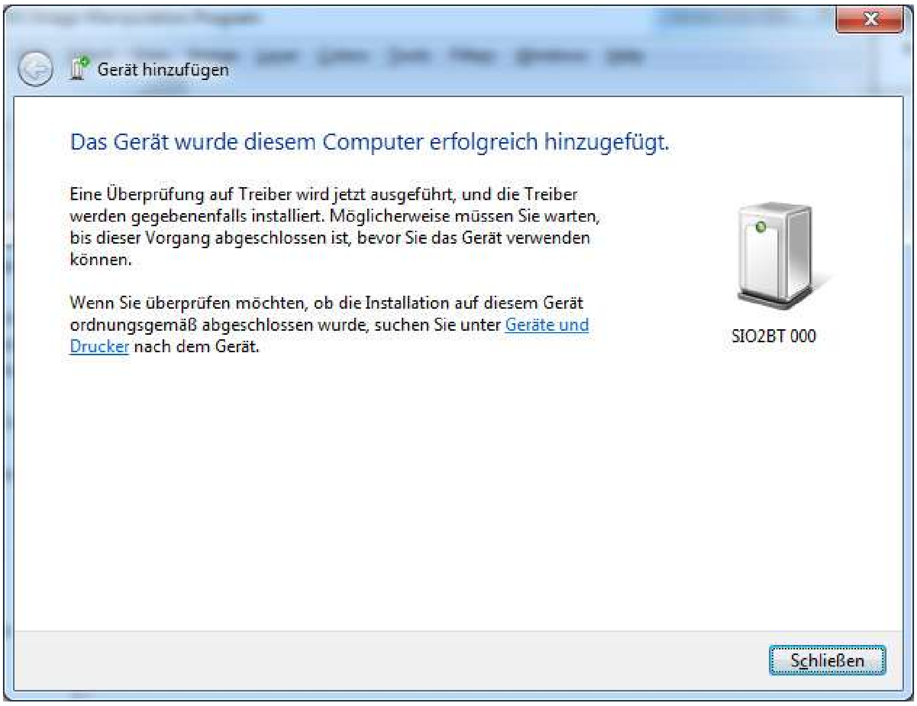
Geben Sie den Kopplungscode für das Gerät ein.

Hiermit wird überprüft, ob Sie die Verbindung mit dem richtigen Gerät herstellen.

1234

Der Code wird entweder auf dem Gerät angezeigt, oder er befindet sich in den im Lieferumfang des Geräts enthaltenen Informationen.

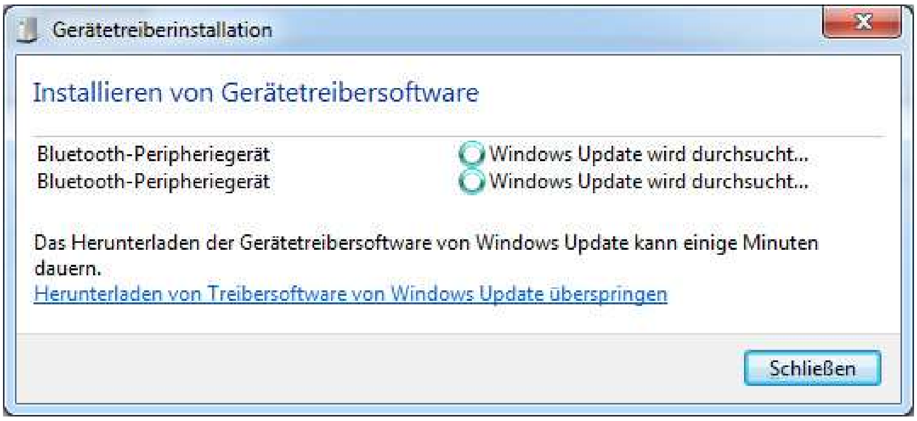
Wie muss vorgegangen werden, wenn der Code für die Gerätekopplung nicht gefunden werden kann?



Das Gerät wurde diesem Computer erfolgreich hinzugefügt.

Eine Überprüfung auf Treiber wird jetzt ausgeführt, und die Treiber werden gegebenenfalls installiert. Möglicherweise müssen Sie warten, bis dieser Vorgang abgeschlossen ist, bevor Sie das Gerät verwenden können.

Wenn Sie überprüfen möchten, ob die Installation auf diesem Gerät ordnungsgemäß abgeschlossen wurde, suchen Sie unter Geräte und Drucker nach dem Gerät.

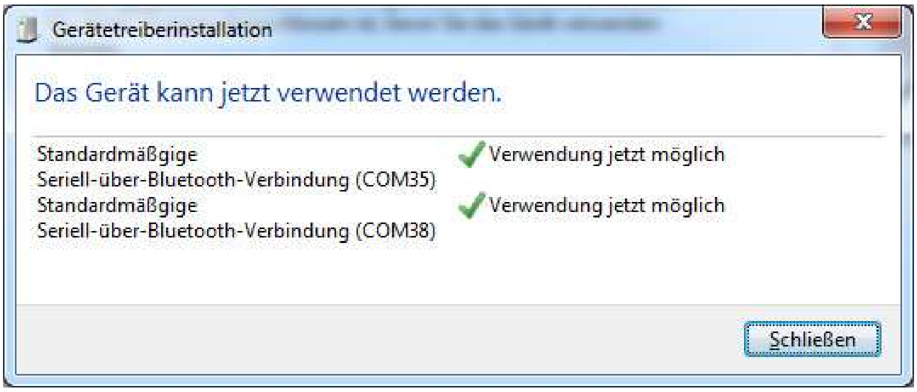


Installieren von Gerätetreibersoftware

Bluetooth-Peripheriegerät	Windows Update wird durchsucht...
Bluetooth-Peripheriegerät	Windows Update wird durchsucht...

Das Herunterladen der Gerätetreibersoftware von Windows Update kann einige Minuten dauern.

Herunterladen von Treibersoftware von Windows Update überspringen



Gerätetreiberinstallation

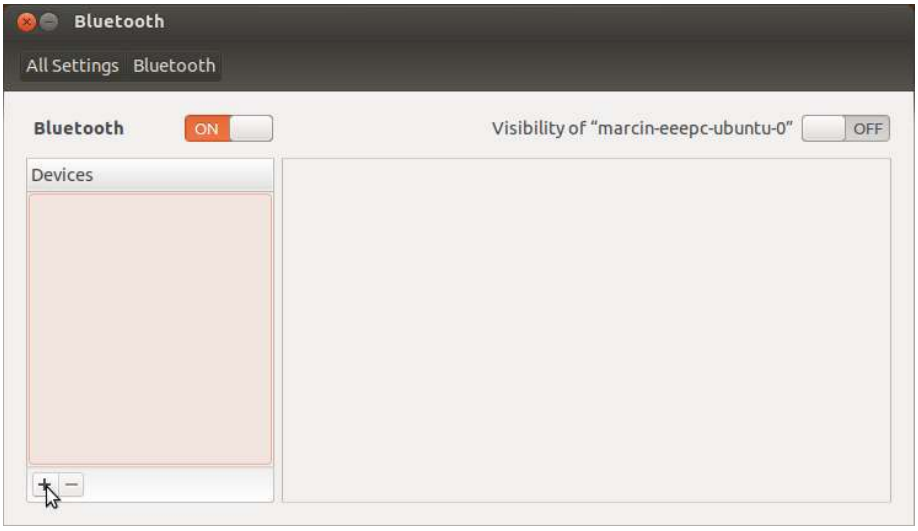
Das Gerät kann jetzt verwendet werden.

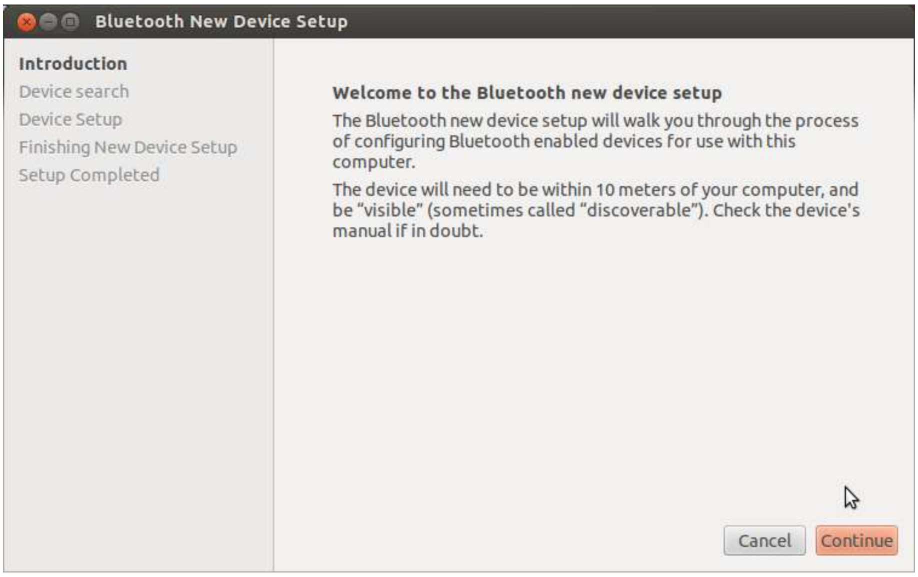
Standardmäßgige Seriell-über-Bluetooth-Verbindung (COM35)	✓ Verwendung jetzt möglich
Standardmäßgige Seriell-über-Bluetooth-Verbindung (COM38)	✓ Verwendung jetzt möglich

Windows automatically creates two virtual serial ports after successful pairing. Only the first of them is useful for SIO2BT.
Please be patient, setting up virtual serial ports takes a few minutes!

Bluetooth under Linux

Pairing process under Ubuntu is presented on the screen shots below:

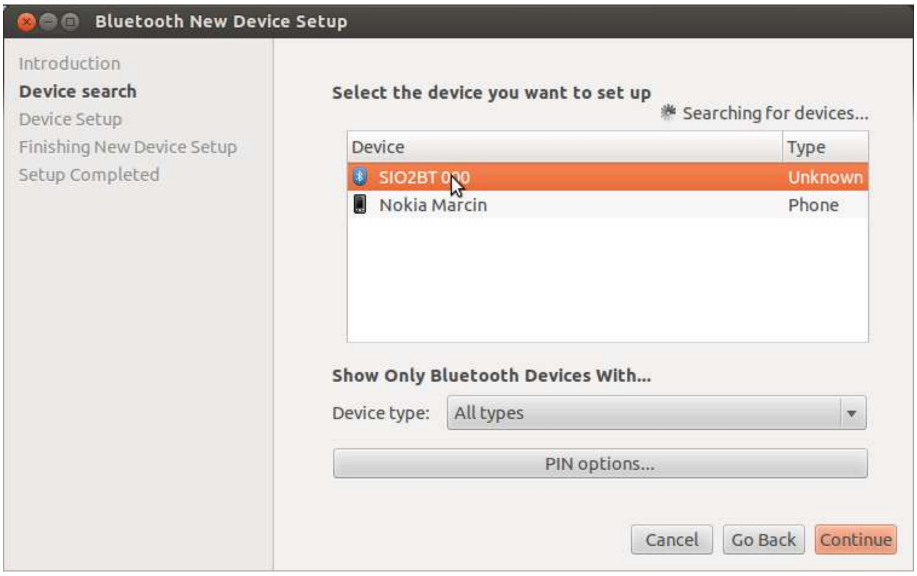




Welcome to the Bluetooth new device setup

The Bluetooth new device setup will walk you through the process of configuring Bluetooth enabled devices for use with this computer.

The device will need to be within 10 meters of your computer, and be “visible” (sometimes called “discoverable”). Check the device's manual if in doubt.



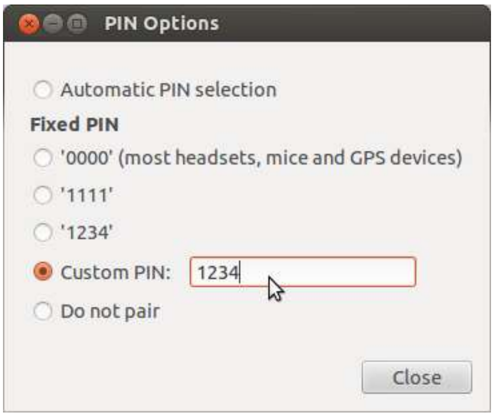
Select the device you want to set up

Searching for devices...

Device	Type
SIO2BT 000	Unknown
Nokia Marcin	Phone

Show Only Bluetooth Devices With...

Device type: All types



○ Automatic PIN selection

Fixed PIN

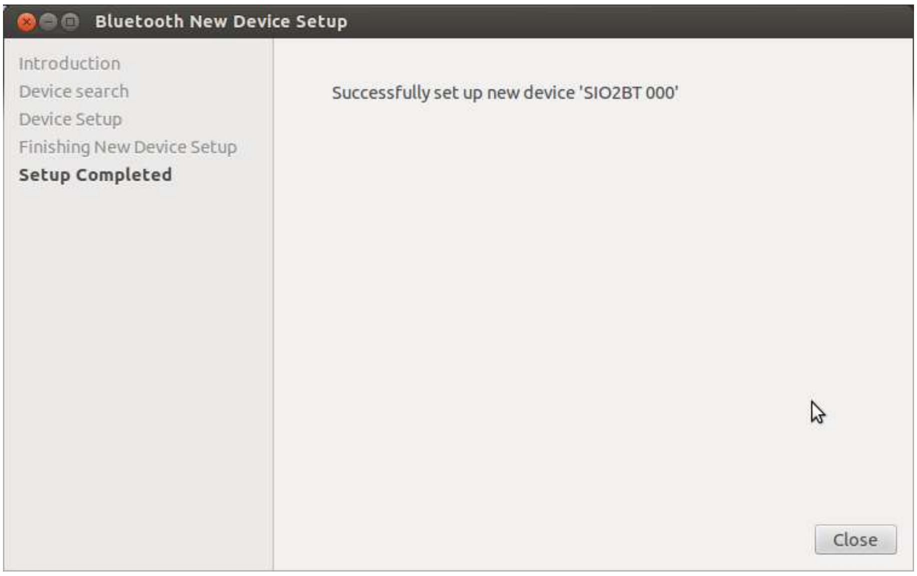
○ '0000' (most headsets, mice and GPS devices)

○ '1111'

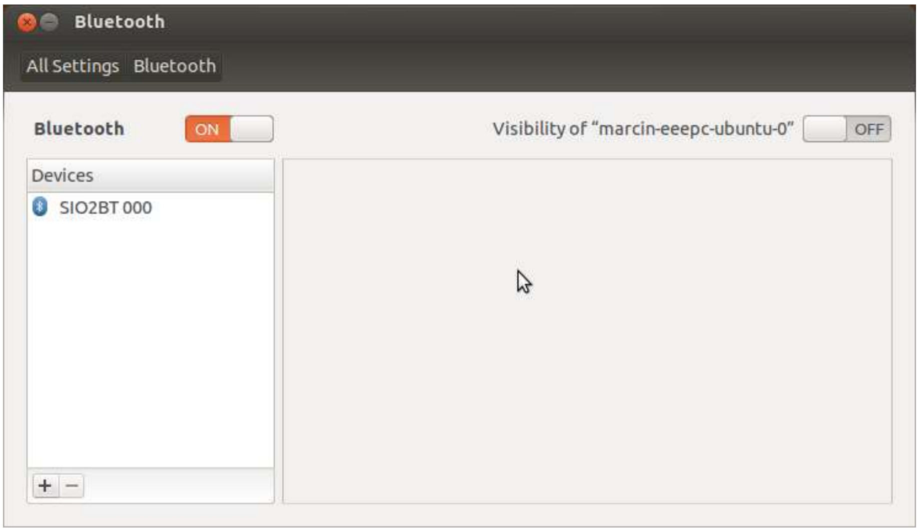
○ '1234'

● Custom PIN: 1234

○ Do not pair



Successfully set up new device 'SIO2BT 000'



Ubuntu does not create virtual serial ports automatically after successful pairing.
Open a terminal and type in the following command:

```
$ sudo hcitool scan
```

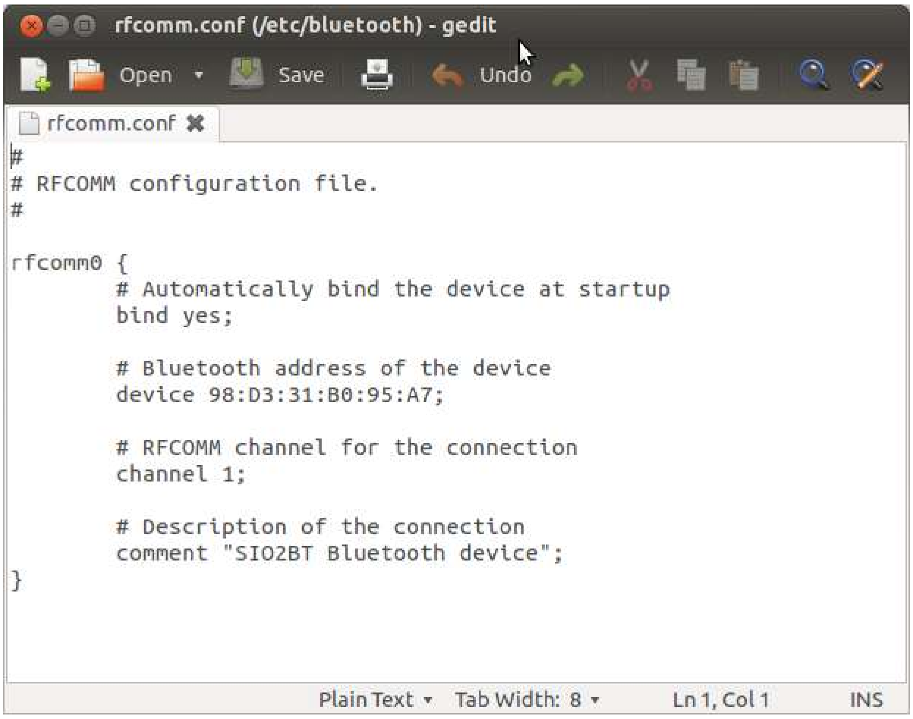
The hcitool searches for discoverable devices and displays their Bluetooth addresses and Bluetooth friendly names:

```
Scanning ...

98:D3:31:B0:95:A7 SIO2BT 000
```

You can create a virtual serial port for your SIO2BT by typing:

```
$ sudo rfcomm bind /dev/rfcomm0 98:D3:31:B0:95:A7
```

or you can let it happen automatically, by editing `/etc/bluetooth/rfcomm.conf` file:



```
#
# RFCOMM configuration file.
#

rfcomm0 {
        # Automatically bind the device at startup
        bind yes;

        # Bluetooth address of the device
        device 98:D3:31:B0:95:A7;

        # RFCOMM channel for the connection
        channel 1;

        # Description of the connection
        comment "SIO2BT Bluetooth device";
}
```

Note: Every Bluetooth device has a unique address, so you need to replace the example address above with the actual one of your SIO2BT Transceiver.

Your SIO2BT is accessible as `/dev/rfcomm0`

Now add the current user to the `dialout` group to give him access to the `rfcomm0` device:

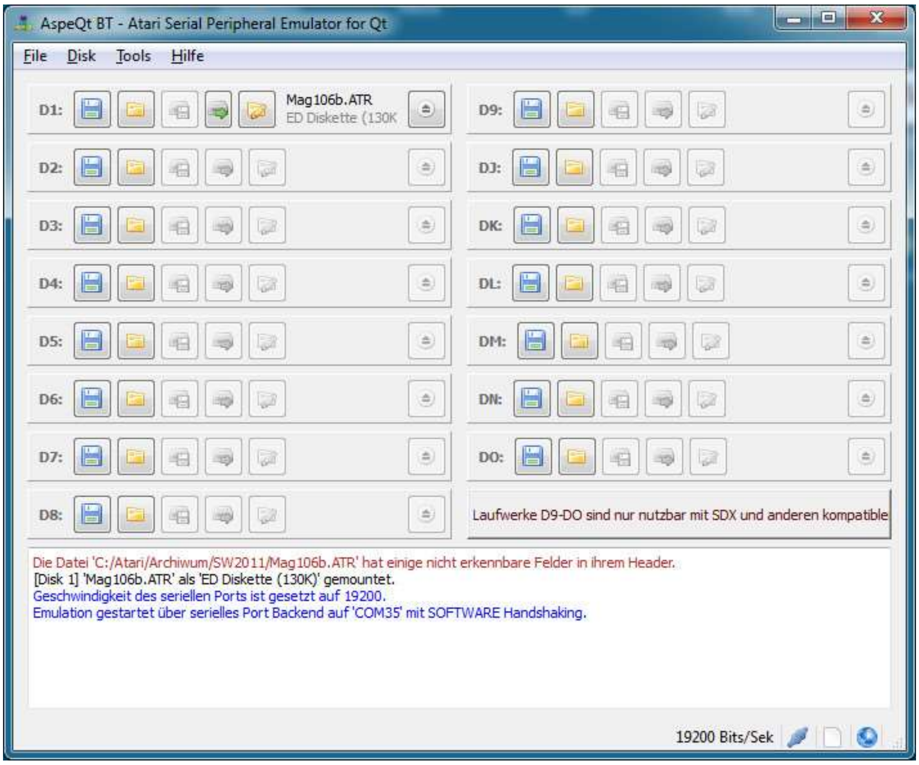
```
sudo adduser $USER dialout
```

Emulating SIO devices with SIO2BSD

SIO2BSD is a PC tool that handles SIO communication under FreeBSD, Linux and MacOS X. Example use: sio2bsd -d10 -s /dev/rfcomm0 image.atr

Emulating SIO devices with RespeQt

The Windows binaries can be downloaded from Github (see links at the end of this document).


AspeQt BT - Atari Serial Peripheral Emulator for Qt
File
Disk
Tools
Hilfe
D1:
Mag106b.ATR
ED Diskette (130K
D2:
D3:
D4:
D5:
D6:
D7:
D8:
D9:
DJ:
DK:
DL:
DM:
DN:
DO:
Laufwerke D9-DO sind nur nutzbar mit SDX und anderen kompatible
19200 Bits/Sek

```
Die Datei 'C:/Atari/Archiwum/SW2011/Mag106b.ATR' hat einige nicht erkennbare Felder in ihrem Header.
[Disk 1] 'Mag106b.ATR' als 'ED Diskette (130K)' gemountet.
Geschwindigkeit des seriellen Ports ist gesetzt auf 19200.
Emulation gestartet über serielles Port Backend auf 'COM35' mit SOFTWARE Handshaking.
```

Linux users have to compile RespeQt first:

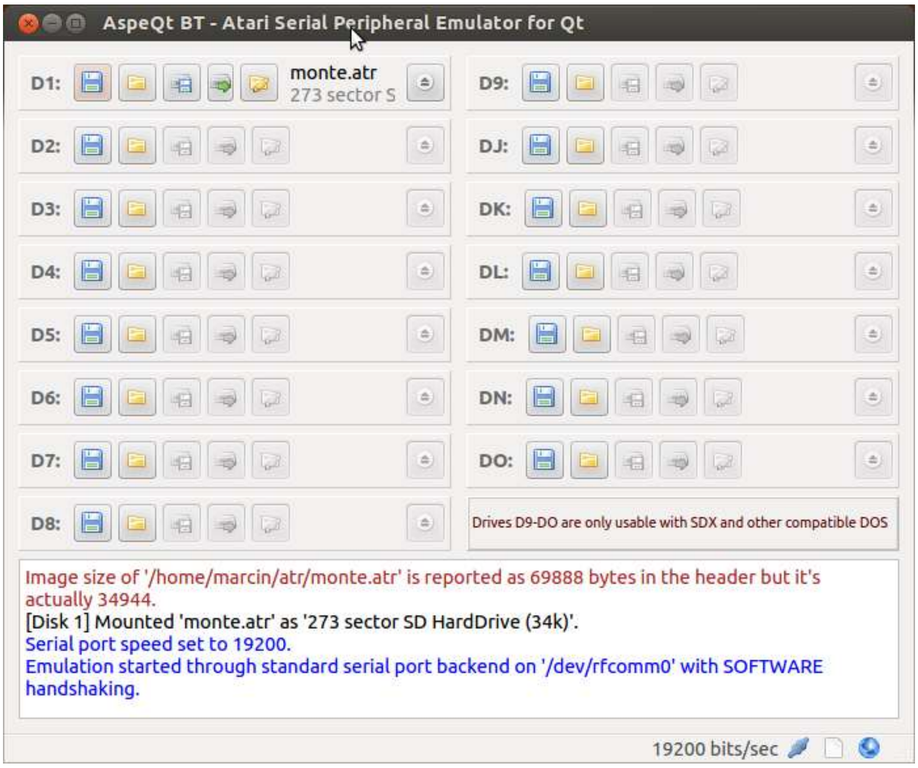
```
$ sudo apt-get install build-essential qt5-default qtbase5-dev
```

```
$ qmake
```

```
$ make
```

And now the tool can be started:

```
$ ./RespeQt
```



D1: monte.atr 273 sector S

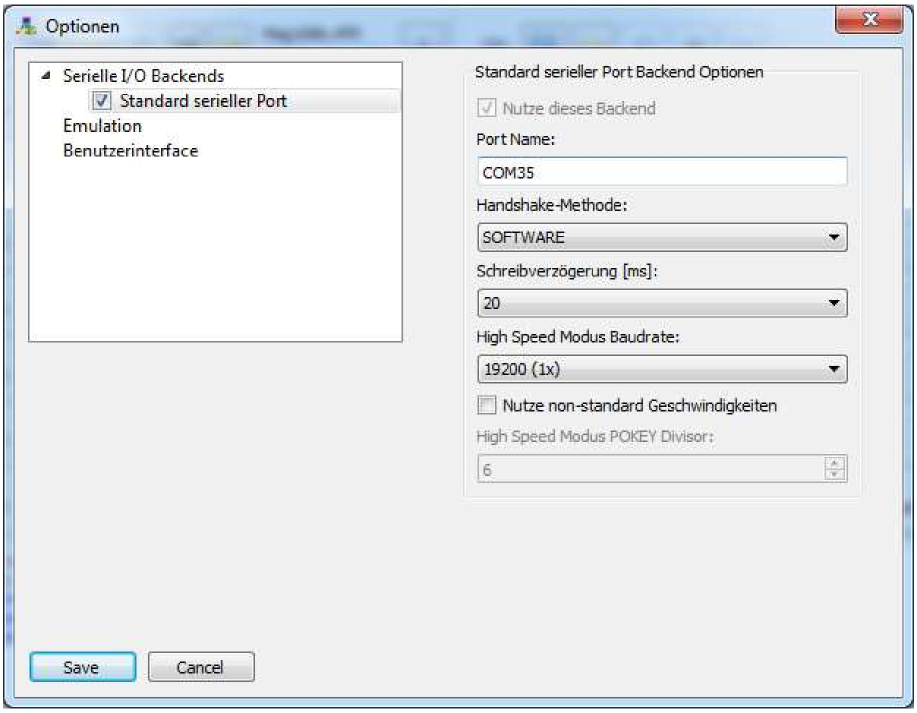
Drives D9-DO are only usable with SDX and other compatible DOS

```
Image size of '/home/marcin/atr/monte.atr' is reported as 69888 bytes in the header but it's
actually 34944.
[Disk 1] Mounted 'monte.atr' as '273 sector SD HardDrive (34k)'.
Serial port speed set to 19200.
Emulation started through standard serial port backend on '/dev/rfcomm0' with SOFTWARE
handshaking.
```

There is a new “SOFTWARE” handshaking method and a new “Write Delay [ms]” parameter introduced. It's default value is 10 milliseconds.

You can increase it if you observe communication problems with the SIO2BT and you can set it to 0 if you want to use this version of RespeQt with serial cables supporting only RxD and TxD.

Selecting SOFTWARE handshaking tells RespeQt to ignore the Command Line status. It processes incoming data from Atari (Data Out) and tries to detect valid command frames.



☑ Standard serieller Port

Standard serieller Port Backend Optionen

☑ Nutze dieses Backend

Port Name: COM35

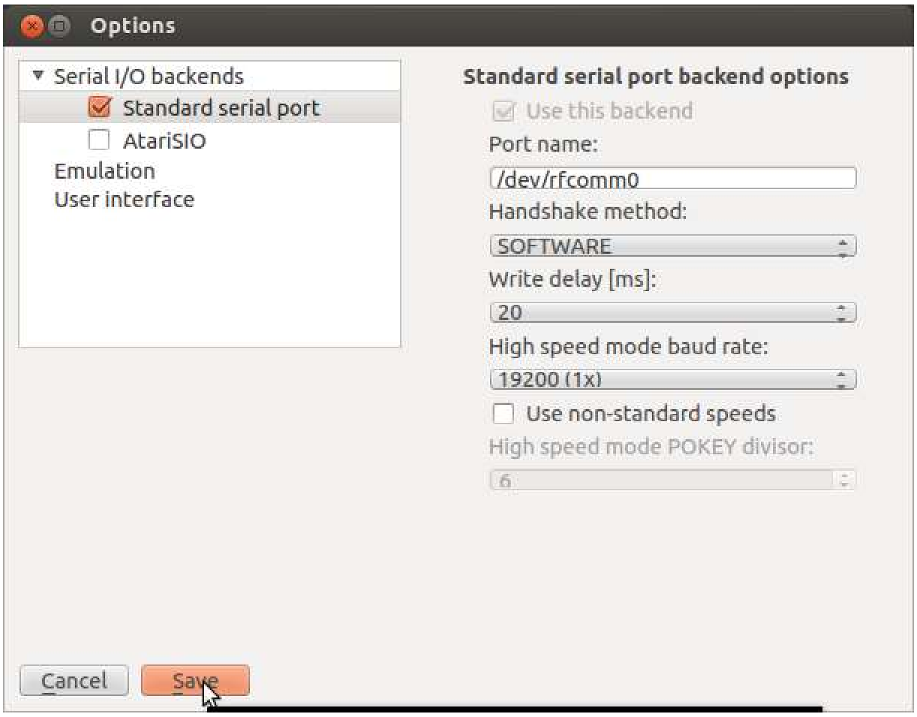
Handshake-Methode: SOFTWARE

Schreibverzögerung [ms]: 20

High Speed Modus Baudrate: 19200 (1x)

☐ Nutze non-standard Geschwindigkeiten

High Speed Modus POKEY Divisor: 6



Standard serial port backend options

☑ Use this backend

Port name: /dev/rfcomm0

Handshake method: SOFTWARE

Write delay [ms]: 20

High speed mode baud rate: 19200 (1x)

☐ Use non-standard speeds

High speed mode POKEY divisor: 6

Raspberry Pi and SIO2BT

The Raspberry Pi Foundation adapted Debian „Jessie“ OS for their small computers.
There are some differences between Ubuntu and Raspbian (this is how Debian for RPI is called).
One of them is, that the Bluetooth software has to be installed on the Raspberry Pi:

```
sudo apt-get install bluetooth bluez blueman
```

The good thing is that you can download precompiled RespeQt binaries from the Github.

You will also need a Bluetooth USB Dongle for Raspberry Pi Zero, 1 or 2.
RPI3 has a Bluetooth Adapter onboard, which works very well with RespeQt and sio2bsd.

Emulating SIO devices with SIO2BT Android App

The Android SIO2BT App can be downloaded from Google Play:

https://play.google.com/store/apps/details?id=org.atari.montezuma.sio2bt

The App emulates up to 4 floppy disks, networking device ($4E) and APE Time Device.

You can select disk images (*.atr), executable files (*.xex, *.com, *.exe) or any other file.

The SIO2BT app uses the XEX Loader from the SDRIVE project.
The files (other than *.atr) are visible to DOS as disks with only one entry (the file itself).
MyDos / SPARTA DOS X users can mount downloaded files and copy them onto the harddrive (for example: KMK IDE2 / Ultimate+SIDE2). This way the flexibility of the Android app can be combined with high performance of the PBI based solutions. DOS 2.0 / 2.5 can handle up to 80KB big files. QMEG, MyDos and SPARTA DOS X can handle files up to 8MiB.

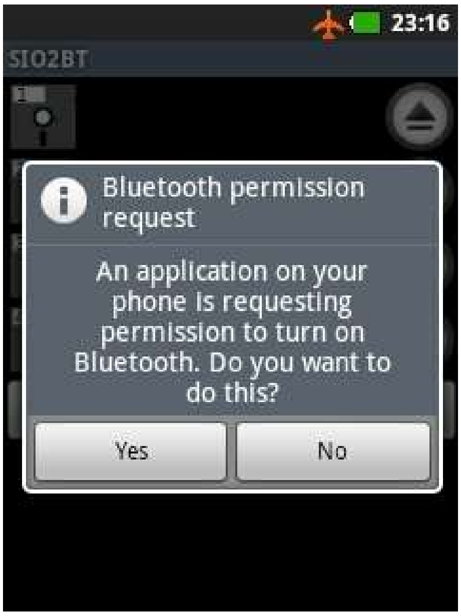
A "long touch" on an executable allows you to choose the xex loader address (default value is

$700). Write protection mode (R/RW) for a disk image can be set with a "long touch", too.
The emulated disks (unless write protected) can be modified (SIO commands: format, write sector, etc. are supported).
In case of communication problems, please increase the delay value in the settings menu.

If the Bluetooth connection is established, it stays alive even if the display is switched off or the user pressed the power button.

Note: SIO2BT Android App works only with bundled hardware (please contact montezuma@abbuc.de for details)

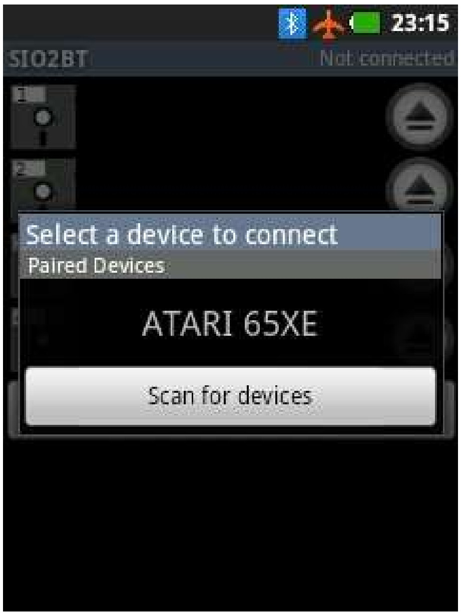
The following screenshots demonstrate briefly how to use the app.



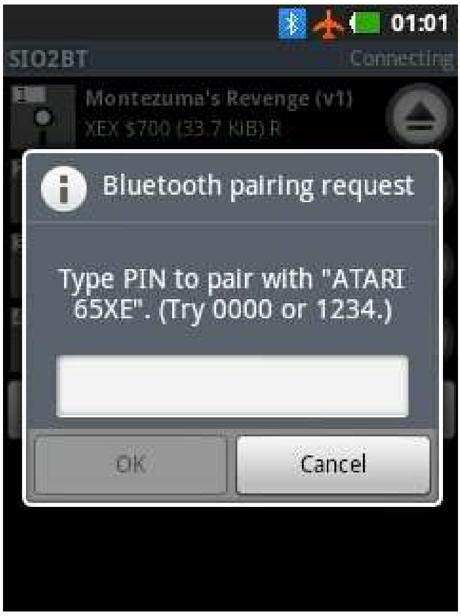
Bluetooth permission request

An application on your phone is requesting permission to turn on Bluetooth. Do you want to do this?

If you have not enabled Bluetooth on your Android device, before starting SIO2BT App, you will be asked to do so.

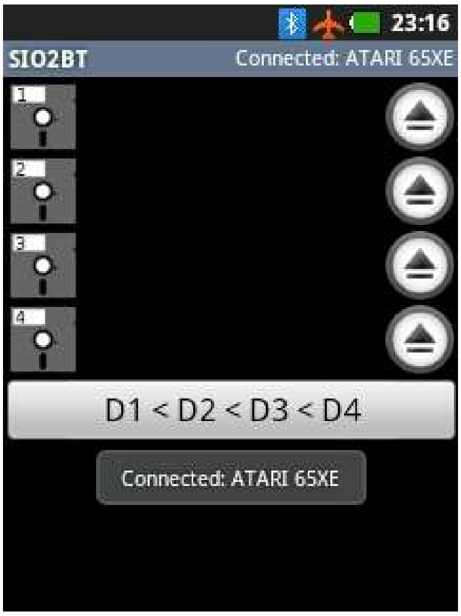


The App presents a list of paired SIO2BT devices.
You may select one of those devices or touch „Scan for devices“ to search for and pair another SIO2BT device.



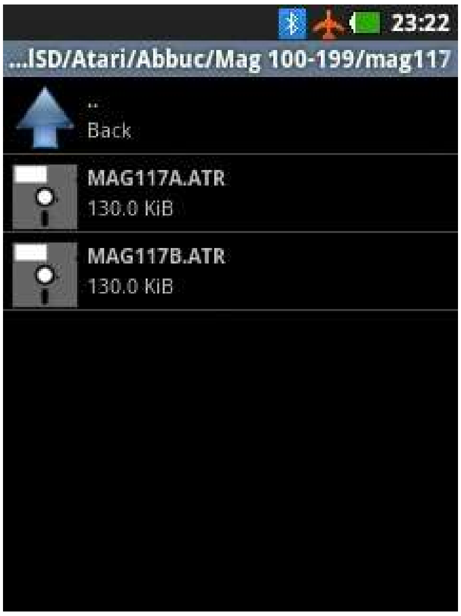
Type PIN to pair with "ATARI 65XE". (Try 0000 or 1234.)

In case of a new SIO2BT device you will be prompted to enter the PIN code.



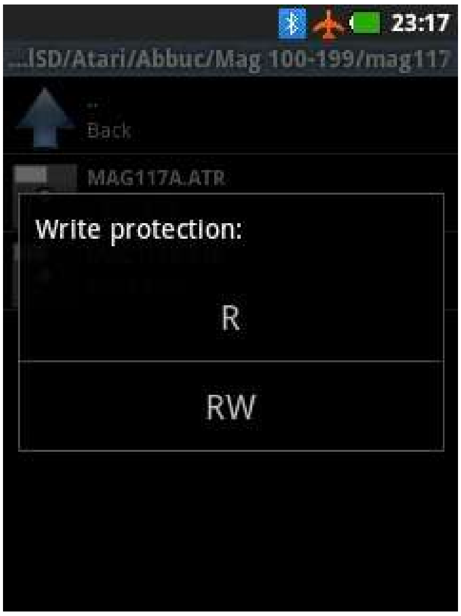
Connected: ATARI 65XE

The connection status is shown in the right upper corner. Touching a disk icon or the place between the disk and eject icons starts the file selection activity.

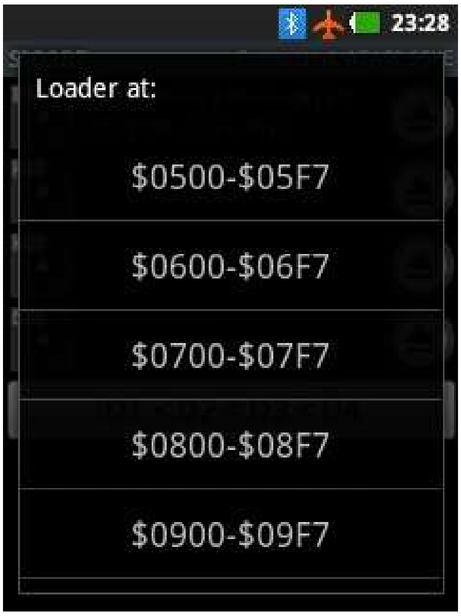


You may navigate through the file system by touching the „back“ icon or by touching directory names to any directory containing Atari files.

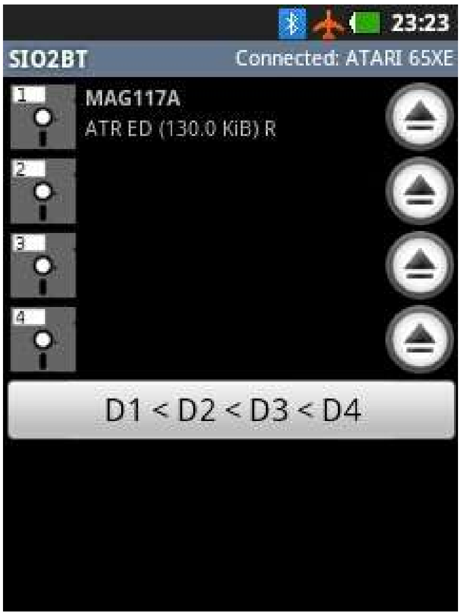
Touching (or long-touching) an ATR or XEX selects the file for mounting and saves the current directory information.



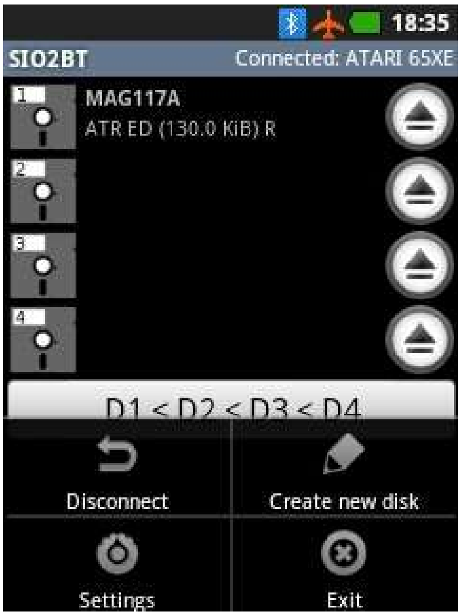
Long-touching an ATR file allows to select the access mode (R - read only, RW - read/write).



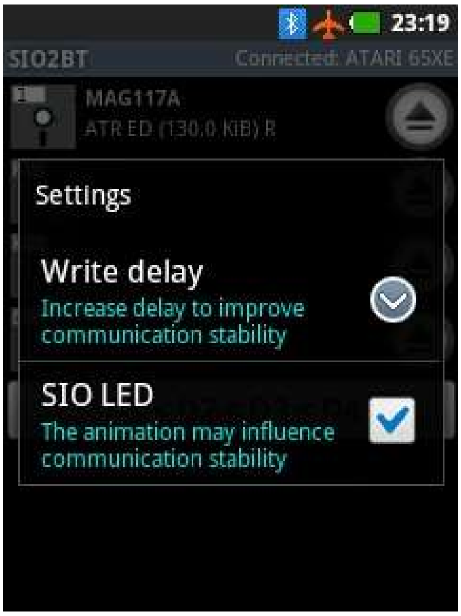
Long-touching a XEX file allows to select the loader address.



The SIO2BT is ready to work.
The button at the button of the screen can be used to quickly swap the disks (useful for multidisk games).



The App Menu has 4 items:
- “Connect” / “Disconnect” (depending on the current connection state) allows you to manage Bluetooth connections to SIO2BT devices
- “Create new disk” creates a DOS 2.5 formatted single density ATR file in the current directory
- “Settings” opens settings menu
- “Exit” closes current Bluetooth connection and exits the app

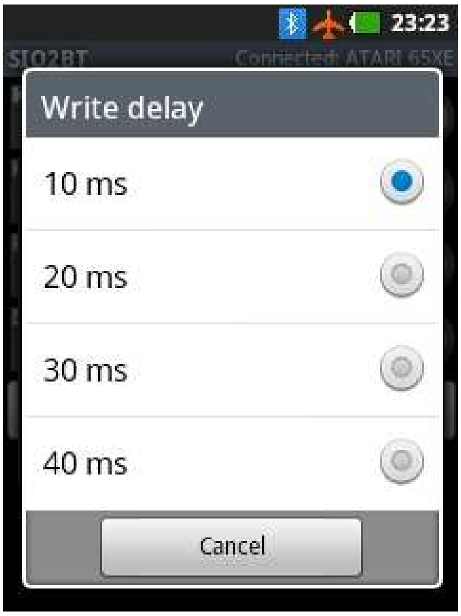


Settings

Write delay
Increase delay to improve communication stability

SIO LED ☑
The animation may influence communication stability

Settings menu has following items:
- “Write Delay” - delay to improve communication stability
- “SIO LED” - SIO visualization
- ”Time” - SIO Clock Device
- ”Network” - SIO Networking Device
- “OPEN Timeout” - timeout for network OPEN operation
- ”READ/WRITE Timeout” - timeout for network READ/WRITE operation



Write delay

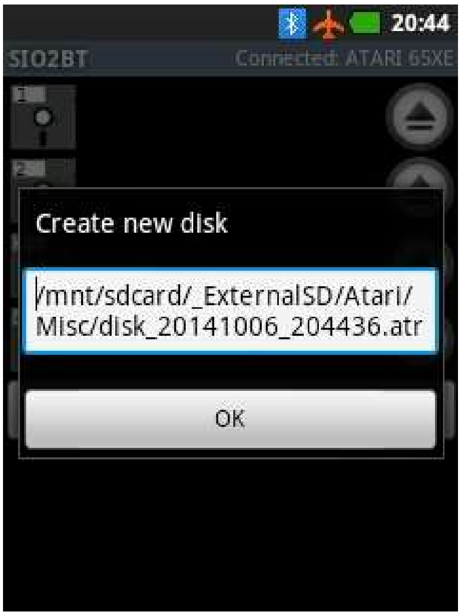
10 ms ●

20 ms ○

30 ms ○

40 ms ○

The bigger the “Write delay”, the more reliable is the SIO communication.



/mnt/sdcard/_ExternalSD/Atari/Misc/disk_20141006_204436.atr

The proposed name for a new ATR file is unique (contains the creation date and time).

Unless you change the path, it will be created in the current directory (directory of the recently mounted file).

The new ATRs are SD DOS 2.5 disks, but they can be formatted by any DOS to others formats.

Emulating SIO devices with AspeQt Android App

The Android AspeQt App can be downloaded from Google Play:

https://play.google.com/store/apps/details?id=org.qtproject.example.AspeQt

In the App settings, you should select SIO2BT device and enter the name of your Bluetooth transceiver. The Bluetooth communication layer was implemented by Greblus and it works very well with all Baudrates supported by the hardware.

Sparta DOS X support for SIO2BT

Sparta DOS X uses it's own SIO procedures per default.
You can tell Sparta DOS X to use the Atari OS procedures (patched for SIO2BT).

To achieve the goal, please create the CONFIG.SYS file on the Sparta DOS X bootable partition. The command below copies the default CONFIG.SYS file to the partition D1.

```
TYPE CAR:CONFIG.SYS >>D1:CONFIG.SYS
```

Edit the file: `ED CONFIG.SYS`

and append `/A` after the line: `DEVICE SIO` (it should like: `DEVICE SIO /A`).
Save the file and restart the Atari.

Since SDX 4.48, the SDX SIO procedures can be configured to support SIO2BT (this is independent of the used OS):

```
SIOSET WAITACK 16
```

increases the timeout and the following command let you set the baudrate for a specific disk:

```
SIOSET X US 8
```

where X is a disk number (for example: 1) and 8 is a hsindex corresponding to 57600.

Note:
If you use Ultimate1MB with SIDE2, you must configure them in the following way:
- SIDE2 switch should be in the XEX loader position
- Ultimate1MB settings:
SPARTADOS X: ENABLED
PBI BIOS: ENABLED
Hard disk: ENABLED

Planetary Defense 2012

Planetary Defense 2012 is a 8-bit ATARI computer game written by Tom Hudson.
The game is included in the Android version of the Atari800 emulator “Colleen” and it uses two special features of the emulator, touch events and high score upload using a “new” B: (browser) device.

Due to the fact that the game uses a high level CIO system, it was easily possible to write a B: handler, which sends the URL over SIO to a Smart Device (emulated in the SIO2BT app).

The pd2012.atr disk images includes a B: device handler and the game itself (thanks to a kind permission from Tom Hudson).

If you enable "Smart Device" in the SIO2BT Android app settings, you can enjoy hi-score submission feature with a real ATARI hardware.

References

Bluetooth Essentials for Programmers, Albert S. Huang, ISBN-13: 978-0521703758

http://en.wikipedia.org/wiki/Bluetooth

https://www.bluetooth.org

http://wiki.pinguino.cc/index.php/SPP_Bluetooth_Modules

http://atariage.com/forums/topic/201133-os-source-code-all-revisions/ (OS source code)

http://atari8.co.uk/ (uFlash, Ultimate 1MB)

http://www.lotharek.pl/product.php?pid=67 (Lotharek's store: Ultimate 1MB)

http://www.atarimax.com/flashcart/forum/viewtopic.php?f=17&t=1585 (MyIDE II forum)

http://www.mr-atari.com/MyBIOS (MyIDE II firmware)

http://www.atarimax.com/myide/documentation (Atarimax store: MyIDE II)

https://github.com/jzatarski/RespeQt/releases (RespeQt)

https://github.com/TheMontezuma/SIO2BSD/releases (sio2bsd)

http://raster.infos.cz/atari/hw/sdrive/sdriveen.htm (SDRIVE)

http://sdx.atari8.info (Sparta DOS X)

https://play.google.com/store/apps/details?id=org.atari.montezuma.sio2bt (SIO2BT App)

https://play.google.com/store/apps/details?id=org.qtproject.example.AspeQt (AspeQt App)

http://xxl.atari.pl/ (XBIOS)

http://blog.3b2.sk/igi/post/Vymena-ATARI-OS-1_2-Change-ATARI-OS-1_2.aspx (OS switch)

http://analog.klanky.com/8bit.htm (Planetary Defense 2012)