miniduino

Manual

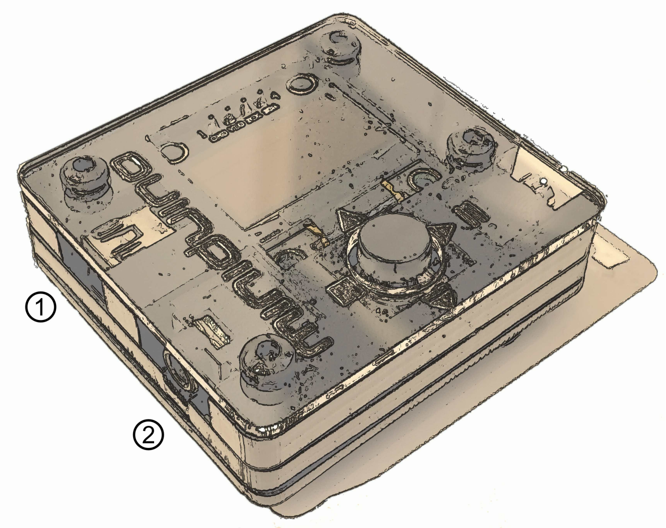
Index

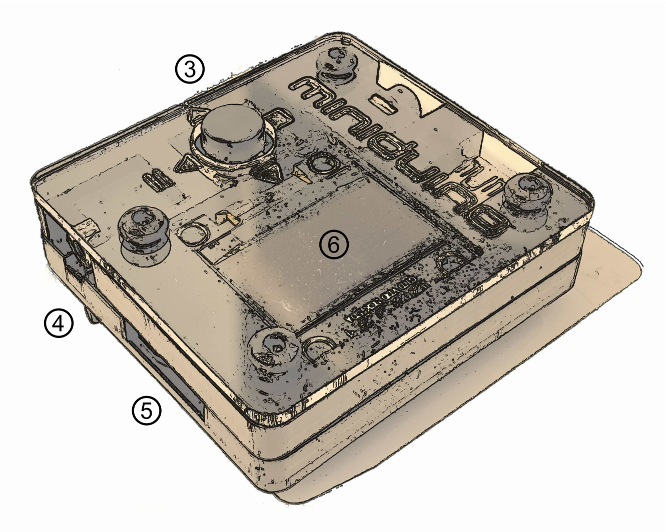
Introduction

Miniduino is a tape file audio player, based on a STM32F103C8T6 microcontroller or ATMega38P, and Maxduino firmware preinstalled.

Maxduino plays, in a very similar way to how cassette tape players worked, digital tape files in format as like `TAP` and `TZX` (ZX Spectrum), `O` (ZX80), `P` (ZX81), `CDT` (Amstrad CPC), `CAS`(MSX) `TSX` (MSX, Acorn, etc). It is also possible to play AY music files as if they were tapes, in order to load the from SpecAY in a ZX Spectrum.

Ports and buttons





1	Power
2	Audio output
3	Control button
4	Motor control
5	SD card slot
6	Screen

Configuration

A SD is needed in order to store the tape files to play. Fast cards (Class 10 or greater) aren't recommended because there can be problems while reading the data. High capacity (SDXC or greater) cards aren't recommended too.

The card must have the first partition formatted as FAT16 or FAT32.

Besides the card, you have to plug an appropriate audio cable to ZXUNO+ audio input. It must have a 3.5 mm stereo jack on one side, and two mono output on the other side (one for each audio channel). The right audio mono is connected to the Miniduino.

If you have a device that can use motor control, you can also use a cable with a 2.6 mm jack.

Copy the tape files (`TAP`, `TZX`, `O`, `P`, `CAS`, `TSX`, etc) to the first partition of the SD card. They can be organized using folders or directories.

> The player shows file and directory entries in the order stored in the internal FAT table, and not alphabetically. If you want to see them ordered, you have to reorder the SD card structure with a utility like Fat Sorter for Windows, FATsort for Linux and MacOS, YAFS, SDSorter or other.

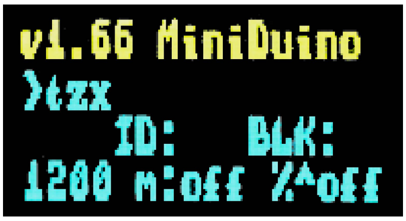
Use

Once the SD card with the data files is inserted, it's turned on plugging in the included USB power cable.

Pressing down the control button shows the options menu which can set the following:

- Baud Rate: Configures turbo speed baud rates when playing 4B blocks in MSX files (`CAS` and `TSX`)
- Motor Ctrl: Enable this option when a control cable is connected to a proper device (Amstrad, CPC, MSX, etc.)
- Converter (TSXCzxpUEFWS): Enables turbo loading `.CAS` and `.TSX` files, changes signal for Spectrum and Amstrad CPC files and/or change parity when playing Acorn Electron and BBC Micro `.UEF` files
- (Skip BLK)): To disable (Skip ON) or enable automatic pause when 2A blocks are found

When not inside the options menu, the control button is used as a four directional control joystick, which has two different behaviours depending whether the player is stopped or paused.

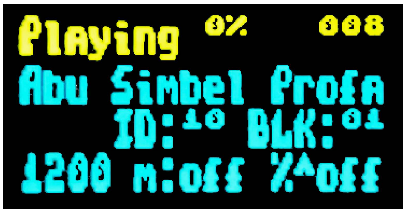


```
v1.66 MiniDuino
>tzx
     ID:    BLK:
1200 m:off %^off
```

When the player is stopped (file and directories browser):

- Up and Down move through the current files and directories list
- Left (Stop) goes one level up in the directory tree
- Right (Play/Pause) enters into a directory or, if the selection is a file, tries to play it

Once a file is being played, stop playing it with the left button (Stop) or pause using the right button (Play/Pause).



```
Playing 0%    008
Abu Simbel Profa
    ID:10 BLK:01
1200 m:off %^off
```

When in pause (tape block browser:

- Up and Down mov through the tape block files already played (useful for multiload titles, to load a previous level block, for example)
- Left (Stop) cancels the player and goes back to file and directory browser mode
- Right (Play/Pause) continues playing from the selected block
- Press down the control button to enable or disable turbo mode for MSX

Making TZX or TSX files from other formats

While there are same tape file formats (Commodore, Camputers Lynx, etc.) not supported by Maxduino, there are some programs that can, more or less successfully, embed audio data in a TSX or TZX file, which then can be used with Miniduino.

MakeTSX

You can use the following command with NataliaPC's MakeTSX to create a TSX file with embedded audio:

```
...MakeTSX -b15 -wav audio_file.wav -tsx new_file.tsx
```

RetroConverter

Jorge Fuertes RetroConverter can create a TZX file with a command like this:

```
...retroconv audio_file.wav new_file.tzx
```

Maxduino firmware upgrade

Maxduino firmware is periodically updated and improved. You can track the changes and improvements either at the Va de Retro forums or at the GitHub project page. To take advantage of this improvements, the Miniduino flash image must be flashed with the updated firmware version.

There are two Miniduino models, one based on STM32 microcontroller, and another one based on ATMega328P.

STM32 Model

Environment setup

Firmware flashing is done from a computer (Windows, Mac, Linux) with Arduino IDE installed.

You have to install SDFat (1.1.4) software library selecting the menu option Program → include library → manage libraries

Minidiuno microcontroller support must also be added. This is done in two steps:

First, adding ARM Cortex M3 support from menu Tools → board → board manager, and installing "Arduino SAM boards (Cortex-M3)"

Then, you have to add STM32 microcontroller support, downloading the file available at this link.

Extract the contents to the current user directory in

```
...Arduino/hardware/Arduino_STM32
```

If on Windows, install the USB device controller, running (with elevated privileges:

```
 ...\drivers\win\install_drivers.bat
```

On Linux, install with root privileges the necessary `udev` rules:

```
...tools/linux/install.sh
```

On MacOS, if Miniduino does not appear as USB device in Arduino ID when plugged, it may be necessary to install libusb.

Finally, when on Mac or Linux, the file `maple_upload` inside `Arduino_STM32` has to be changed with a text editor. Those lines do not work:

```
if [ $# -eq 5 ]; then
    dfuse_addr="--dfuse-address $5"
else
    dfuse_addr=""
fi
```

And they have to be changed into this:

```
dfuse_addr=""
```

Upgrade

Once you have the environment ready, download the software from the official repository in GitHub

> ℹ Miniduino player with STM32 microcontroller is only supported from 1.65 and up

Load the project file with Arduino IDE (for example `MaxDuino_v1.69.ino`).

Check in the file `userSTM32Config.h` that all logo entries are commented except for Miniduino and, if not, change them.

```
...
//#define tanque4
//#define tanque1
//#define dostanques
//#define cablemax
//#define sony
#define miniduino
...
```

Connect the Miniduino device to the computer using the USB cable, and find the assigned port, normally with a name like "Maple Mini" (for example: COM5 Maple Mini)

Set the following options in the "Tools" menu:

```
Board: Generic STM32F103C Series
Variant: STM32F103C8 (20k RAM, 64k Flash)
Upload Method: STM32duino bootloader
CPU Speed: 72Mhz (Normal)
Optimize: Smallest (default)
Port: <Previously identified port>
```

Finally, click on the firmware load button and wait for a few seconds while the project is compiled and loaded into the device.

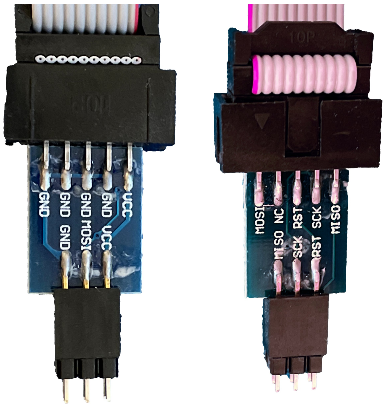
If everything has been done correctly the Miniduino will restart and show on the screen the new firmware version.

ATMega328P Model

Environment setup

Requirements:

- One hex key with the right socket size for the cover screws
- USBasp flash programmer



Also, firmware flashing is done from a computer (Windows, Mac, Linux) with Arduino IDE installed.

You have to install SDFat (1.1.4) software library selecting the menu option Program → include library → manage libraries.

Upgrade

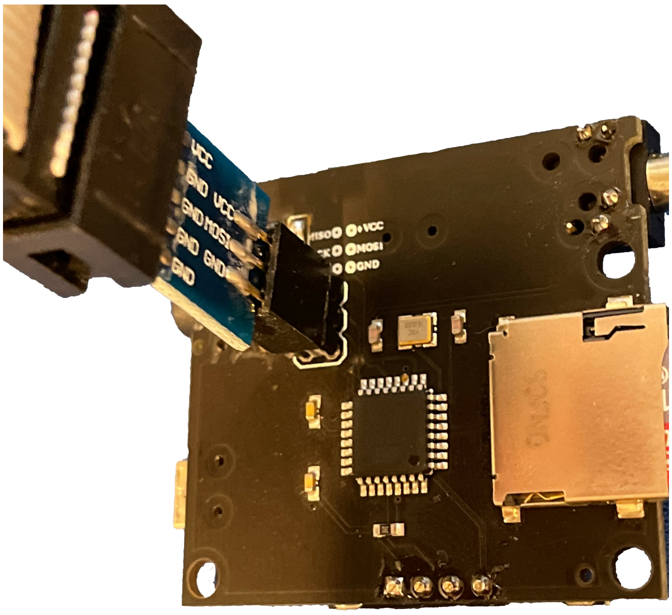
Once you have the environment ready, download the software from the official repository in GitHub

Load the project file with Arduino IDE (for example `MaxDuino_v1.69.ino`).

Check in the file `userconfig.h` that all logo entries are commented except for Miniduino and, if not, change them.

```
...
//#define tanque4
//#define tanque1
//#define dostanques
//#define cablemax
//#define sony
#define miniduino
...
```

Connect the Miniduino device to the USBasp programmer, making sure that the connector is in the right position (i.e VCC with VCC, MOSI with MOSI, GND witch GND, etc.), and connect the USB adapter to the computer



Set the following options in the "Tools" menu:

```
Board: Arduino Pro or Pro Mini
Processor: ATmega328P (5V,16 Mhz)

Programmer: "USBasp"
```

Finally, keep pressed the computer keyboard Shift key while clicking on the firmware load button and wait for a few seconds until the project is compiled and loaded into the device.

If everything has been done correctly the Miniduino will restart and show on the screen the new firmware version.