

8051duino Manual version 0.3 EN

polprog 12.10.22

8051duino is an Arduino-style board based around an 8051 or compatible microcontroller. The board supports 64kB of RAM and a socketed ROM of up to 64kB. Additionally the databus and UART are broken out.

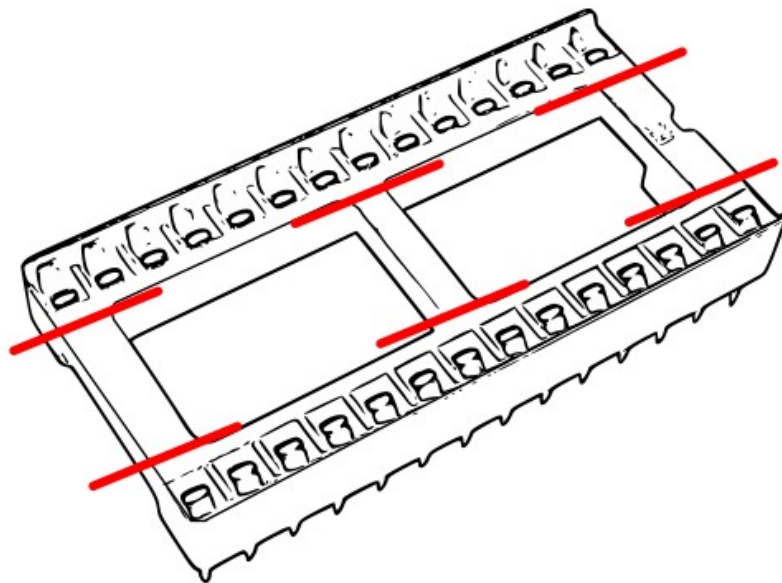
In this version, MCS-52 Basic¹ is included in the external ROM.

PCB errata for rev 2

- Pin 1 of JP1 needs a pullup resistor (10k-50k) to +5V
- Pin 1 of D1 needs to be cut away from ground (on both PCB sides!) and connected to +5V. D1 needs to be installed accordingly (the micro's GPIO pulls it down)
- The crystal should be 11.059MHz (the value is not given, but this plays nicely with the UART)

Assembly information

The RAM chip is directly under the ROM chip. Same goes for the latch under the 8051. The ROM should be installed in a modified socket, that has the inside parts cut out like in the figure below to make room for the RAM.



The MCU can be installed in a regular socket, but please insert both the U1 socket and U3 (latch) before soldering them down. The fit between them is quite tight. This will be fixed in the next revision

Quick start with MCS-52 Basic

Connect the USB-UART to 8051duino and start the terminal program on the PC. The UART pins are in top right corner of the board, looking at it so that the power input is to the left). MCS-52 Basic automatically detects baudrate, so after power up you need to press space several times. If there is garbage on the terminal screen, please reset the board and

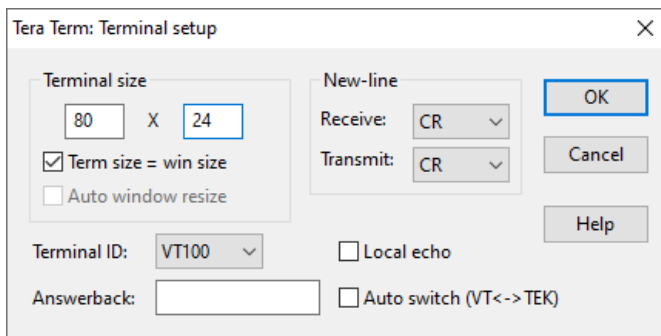
¹ Copyright (c) 1986 Intel Corporation

try again. If the characters you type appear twice, then you need to turn off "Local Echo" in your terminal software.

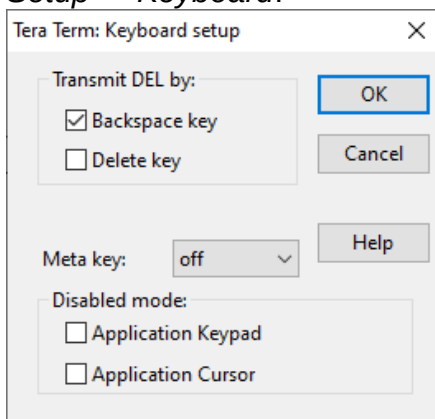
Terminal program configuration:

- Baudrate 9600 8n1
- Local Echo off
- Newline transmits CR
- Backspace sends the backspace character

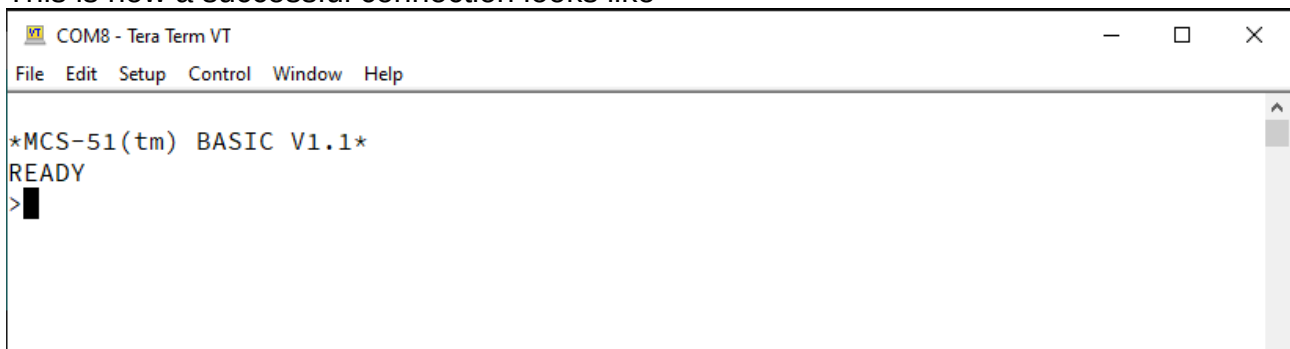
Example TeraTerm setup: Menu *Setup -> Terminal*



Setup -> Keyboard:



This is how a successful connection looks like



First BASIC program

In BASIC environment, programs are entered by giving a set of numbered lines containing instructions. Lines are executed one after another according to their numerical order.

Bold characters should be entered by the user. Example program is below:

```
*MCS-51(tm) BASIC V1.1*
READY
>10 string 110, 10
>20 input "what is your name? ", $(1)
>30 print "Hello, ", $(1)
>LIST
10     STRING 110,10
20     INPUT "what is your name? ",$(1)
30     PRINT "Hello, ",$(1)
```

```
READY
>RUN
```

```
what is your name? chris
Hello, chris
```

```
READY
>
```

Blinking the LED

The LED is on pin P1.0, that is bit 0 of Port 1. The value of the port can be set as in example below:

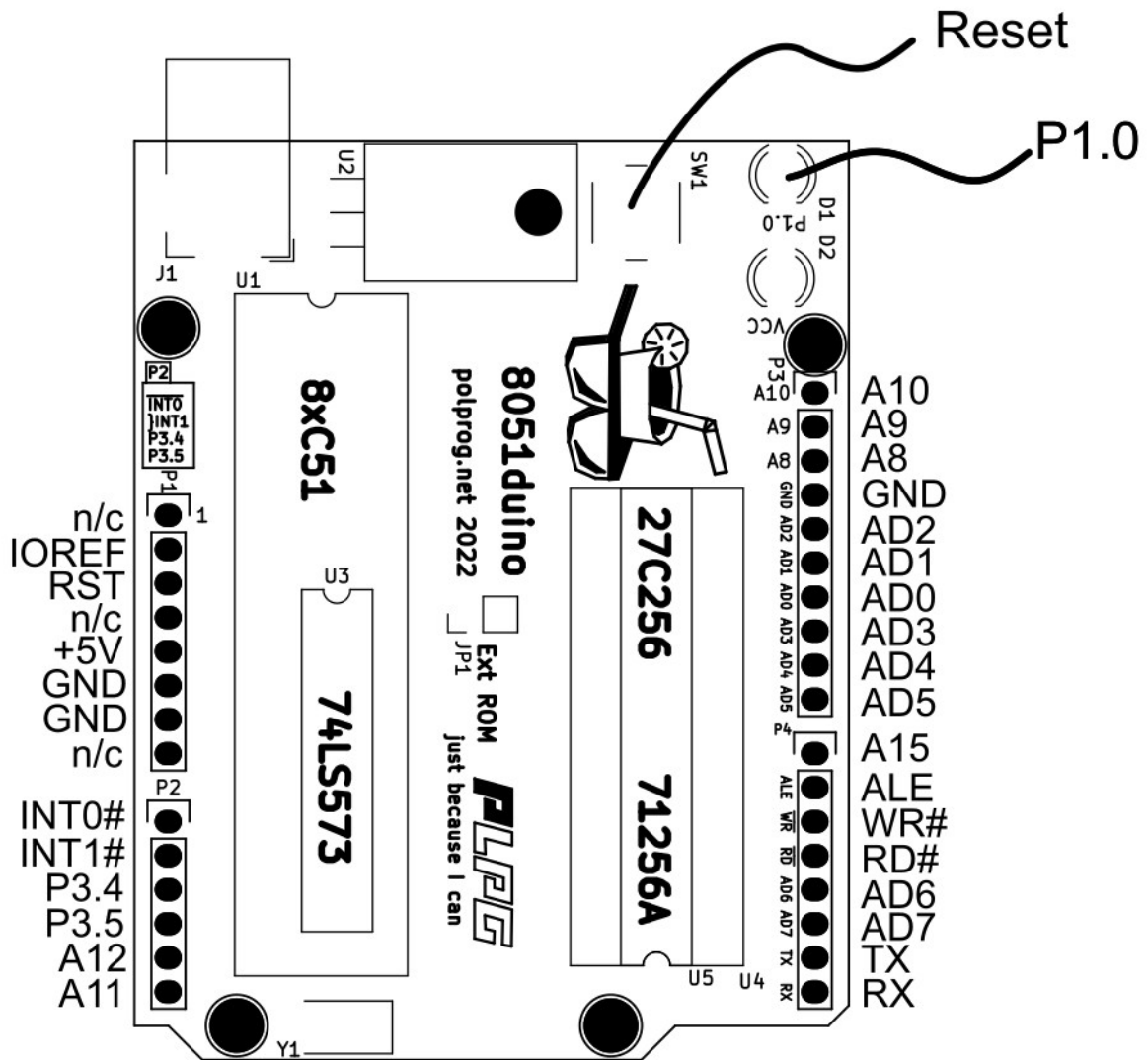
```
>LIST
5     REM BLINK A LED ON P1.0 PROGRAM
10    PORT1=1
20    DO
30    PORT1=PORT1+1
40    FOR A=0 TO 1000 STEP 1
50    NEXT A
60    WHILE 1
```

```
READY
```

You can stop the program execution with Ctrl-C.

Full instructions for MCS-52 Basic is (among other places) available here:
http://www.bitsavers.org/components/intel/8051/MCS_BASIC-52/270010-003_MCS_BASIC-52_Users_Manual_Nov1986.pdf

Board Pinout



Memory Map

ROM 0000-7FFF	External ROM (EA# low, JP1 closed) (when EA# high, JP1 open, depends on 8051 model)
RAM 0000-7FFF	Onboard RAM, A15 = RAM CS#
RAM 8000-87FF	Peripheral IO space (only A0-A10, A15 broken out)

If you are developing in C or asm, please see the schematic as well.

The peripheral connector has AD0-AD7, A8-A10, A15, ALE, WR# and RD# broken out as well as INT0# and INT1#. P3.4 and P3.5 are timer IO.

It's there so you can design your own memory-mapped peripherals such as UARTs, PIOs and other peripherals of the era.

Tools and compatible MCS-51s

For MCS-52 BASIC you don't need much more than a terminal emulator.

For C or asm development you can try the sdcc compiler suite (free) or Keil (paid).

Instead of MCS-52 BASIC you can also burn PAULMON to the ROM. PAULMON is an 8051 monitor program (you want the pm21_2 file, see the memory map).

Or you can have both programs, one in the MCU ROM and one on the external ROM, and use the Ext ROM jumper to switch.

The following 8051 MCUs are known to work:

- Intel P8032AH
- Intel P8052AH
- Intel D87C51
- AT89C52 (Flash ROM)
- Siemens SAB8032A
- Signetics SCN8031

In principle every pin compatible 8051 should work. If you test it with some exotic 8051 MCU, please let me know!

Contact me

If you have any questions, feel free to reach out to me at <gorplop at sdf.org>, polprog on Libera IRC or @polprogpl on Twitter.