VGA and HDMI for ZX Spectrum. Rgb2hdmi by AlexEKB

There are several options for connecting the ZX Spectrum to modern TVs and monitors. Direct connection, without using of additional adapters, is not always possible. Therefore, an active adapter must be used.
Commercially available adapters capture an analog signal, but I wanted to work with a digital one, directly from the Spectrum board. Since the Spectrum has a 4-bit color, we need to find digital signals on the board that corresponds to red (RED), green (GREEN), blue (BLUE) and a high brightness bit (BRT). In models with ULA or CPLD, RGB and BRT signals are often available on these directly. In models implemented on separate logic chips without CPLDs, it is easy to find these signals either. The general procedure of the searching is to get from output to input: if the output is RGB, then you just need to find a place where the RGB digital signals are mixed with a high-brightness bit.

As an example, I will give a part of the scheme of the computer "Leningrad-48". From the emitters of transistors, analog signals, and on the DD36 chip there are their digital prototypes.

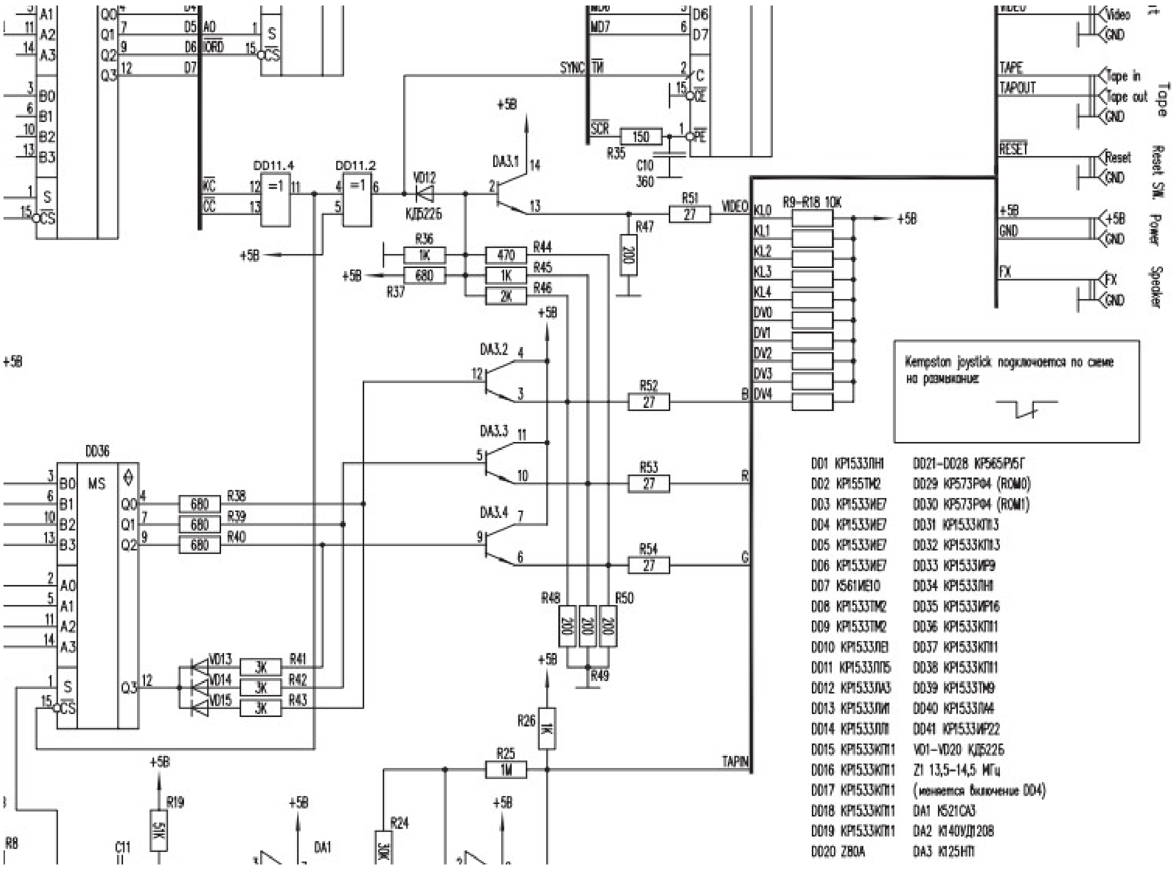


Fig. 1

Here is a part of the ZX Spectrum +2 circuit, where the color component signals are taken from the ULA chip.

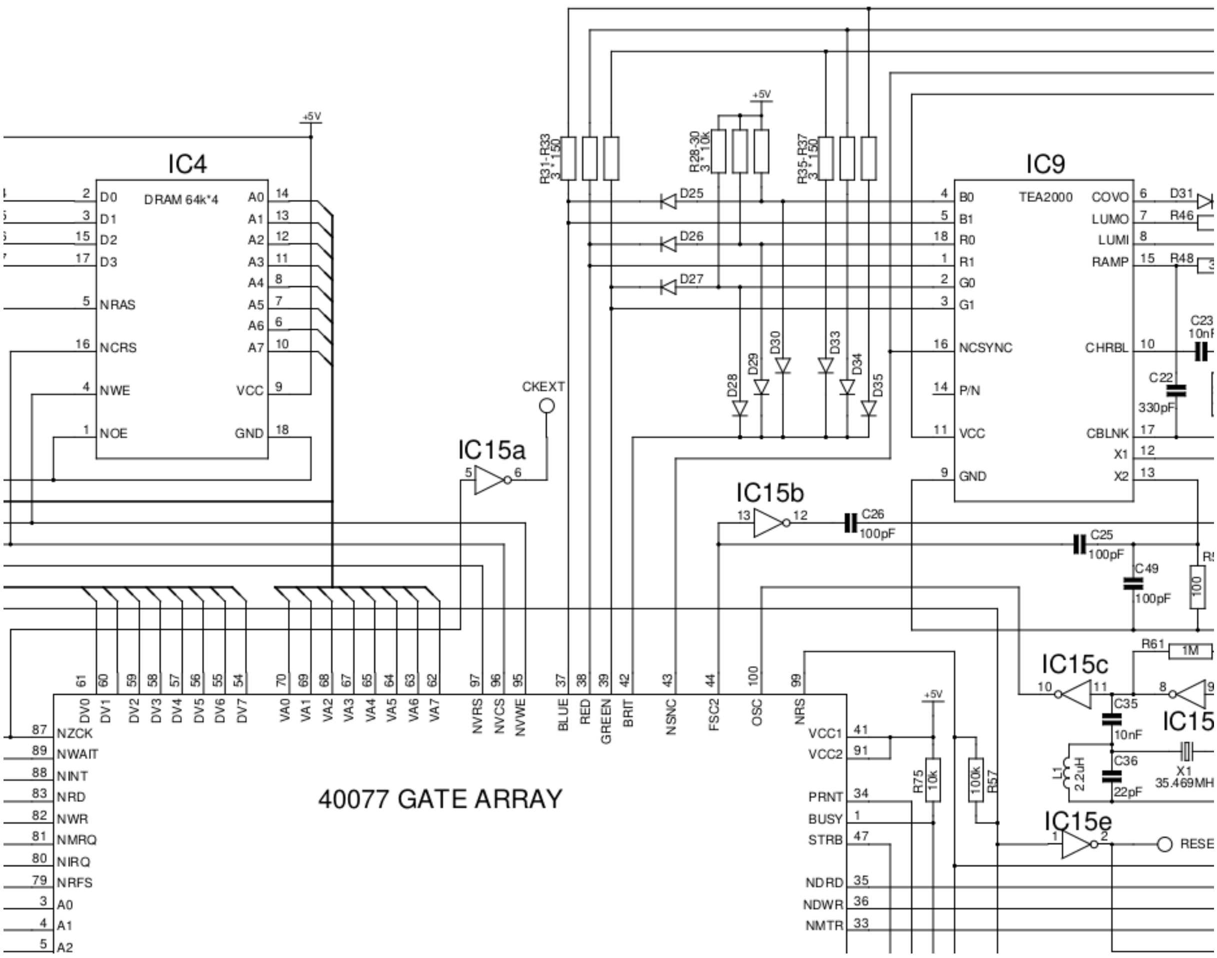
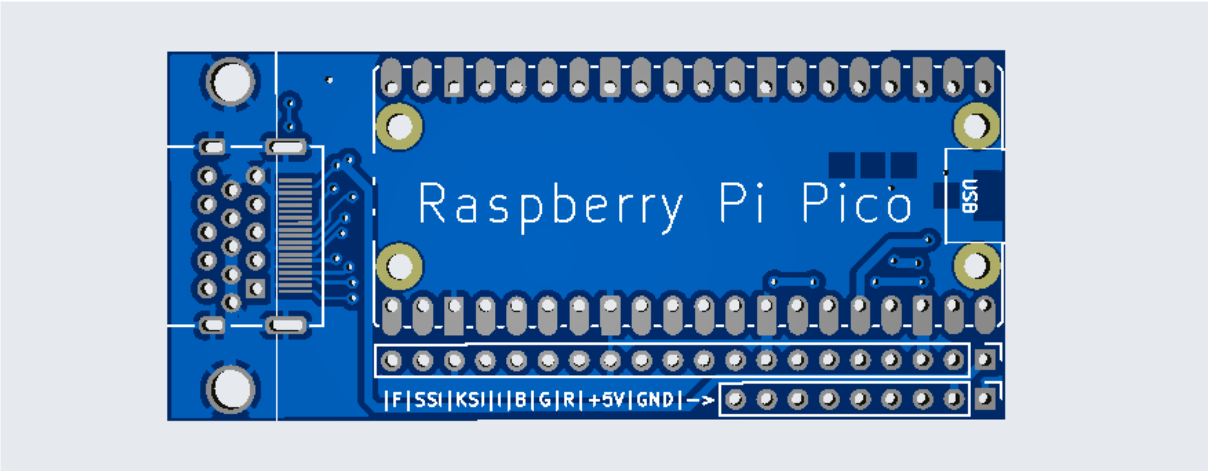


Fig. 2

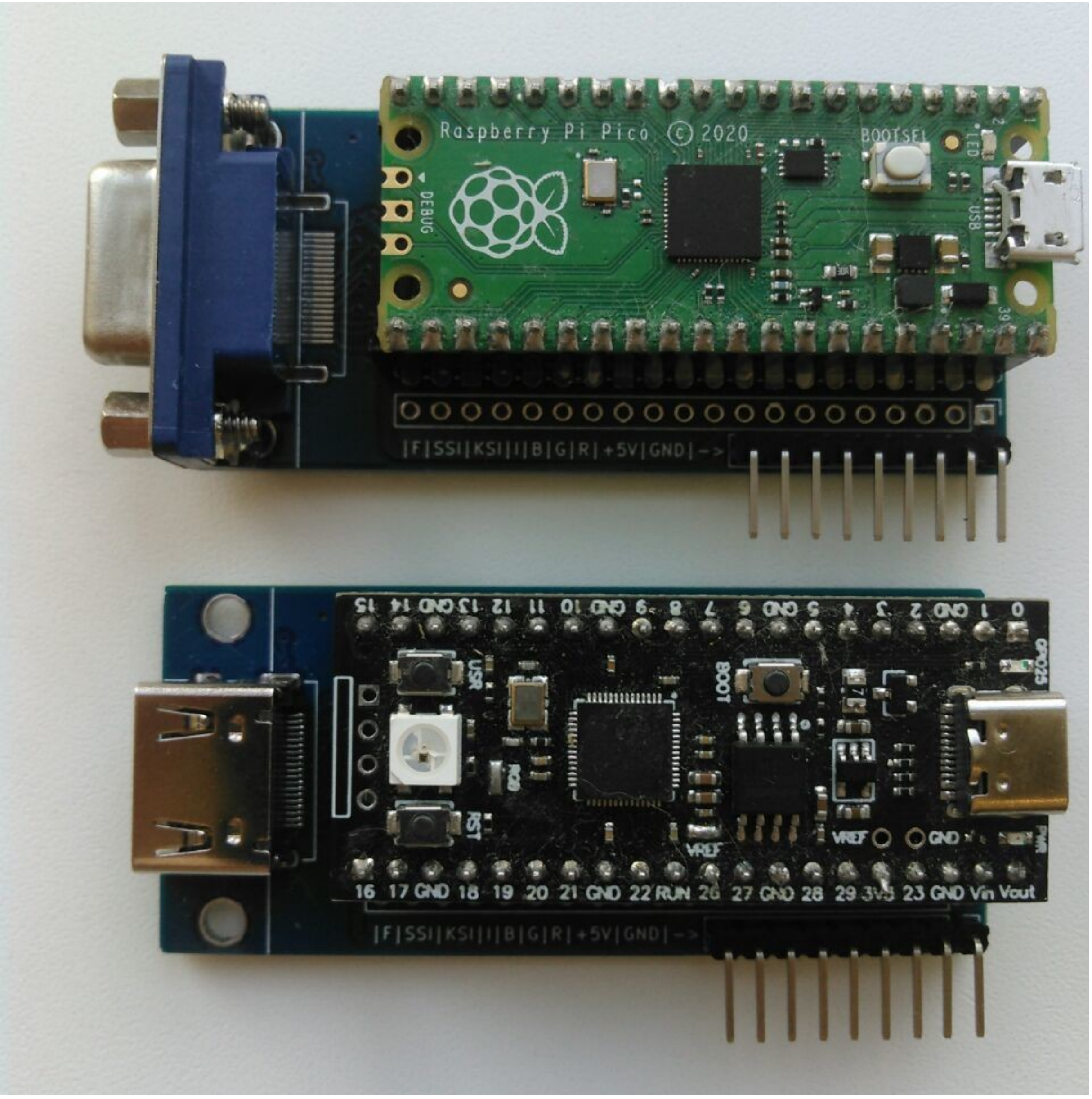
Digital color signals alone are not enough to fully capture an image. Vertical and horizontal sync signals are required. Since the ZX Specrtum computers were designed to connect to TVs, the more traditional way to synchronize is the sync mix. The combined vertical and horizontal sync signals are encoded with different pulse lengths. Vertical sync pulse is longer. On the diagram of Fig. 2 NSNC signal (43 pin of the ULA) produces exactly the sync mix. To simplify the capturing and reduce the amount of data for processing, it would be nice to find another clock signal - the signal pixel clock(P_CLK). In computers such as the ZX Specrtum, pixels are drawn synchronously with processor cycles of 2 pixels per beat. Therefore, you can use the CLK signal from the Z80 processor (6th pin). The capturing of that signal must be done along the signal front and signal recission, to provide the necessary speed of 2 pixels/clock. Such a synchronous capturing can happen while switching of color signals state in the circuit. In order to exclude the influence of transients, the moment of capturing must be shifted by a little offset, relative to the clock signal.

My first experiments on capturing signals and forming a standard VGA signal I carried out on the ESP32 controller. For me was important that the controller stays inexpensive.

Now, I switched to a RP2040 controller in form of a Raspberry Pi Pico board. At the same price as ESP32(~3$)



Pico has the ability to form an image in the form of a DVI signal, which expands the possibilities for connecting to modern monitors. I developed an universal adapter board that allows you to select the video output by changing the firmware, soldering the desired socket (HDMI or VGA) and corresponding resistor values.



When soldering VGA sockets R1-R8 should be 100,100,390,820,390,820,390,820 ohms respectively. For lighter colors, pairs of 390,820 can be replaced with 300, 1000 ohms. Jumpers JP1-JP3 for VGA must be closed.

Software written in C language is divided into VGA, HDMI branches. When compiling, it is possible to select the capture delay, various synchronization methods (mixed sync, separate) and the P_CLK source. With a certain FREQ_CAP, the pixel clock of the selected frequency is formed by the microcontroller itself. In this case, the capture delay is configured by SH_DELAY_CAP. If FREQ_CAP is not defined, then the capture is performed by an external clock signal on the rise and fall. This is usually a signal from the 6th leg of the Z80 processor. The delay in this case is determined by the rise and fall in the constants RISE_CAP_DELAY, FALL_CAP_DELAY. Settings are selected empirically. Several compiled firmware below.