**Pico Elliott 920M Paper Tape Station Emulator**

**=============================================**

**Andrew Herbert – Last Revised 19/11/2023**

**Aim**

**----**

**To build an emulator for the Elliott 920M avionics computer Paper Tape Station (PTS) using a Raspberry Pi Pico.**

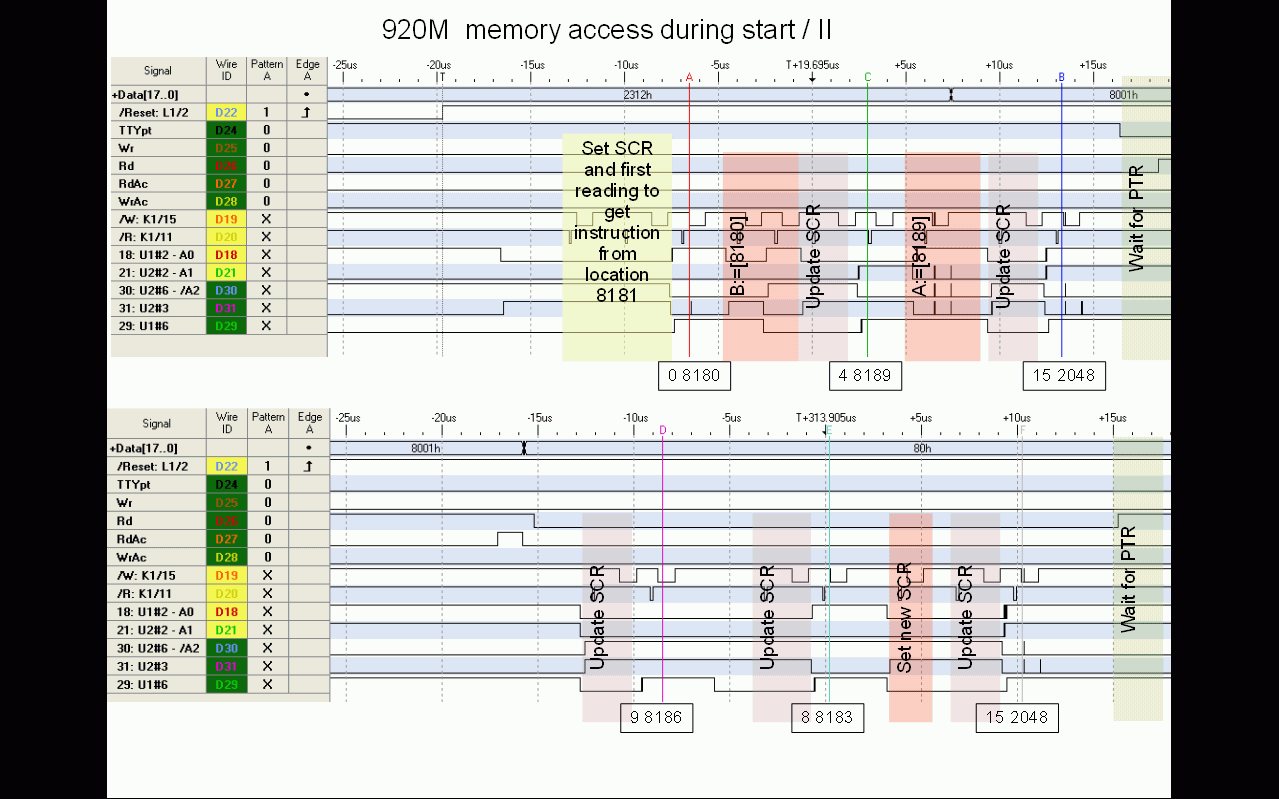
**Principles of Operation**

**-----------------------**

**The software emulates the operation of a paper tape station comprising a paper tape reader, paper tape punch and teletype. The emulation is based on the analysis of the electrical interface of an original 920M by Erik Baiger and Terry Froggatt.**

**The emulation uses Pico GPIO pins to emulate the electronic interface of the 920M PTS. (See hardware section for definition of signals and associated GPIO pins.)**

**920M to Paper Tape Station Interface**

****

**The emulator is initially quiescent, awaiting either a read request (RDRREQ) or punch request (PUNREQ).**

**To make a read request, the 920M raises the signal RDRREQ and awaits a corresponding ACK signal from the PTS. If the request is for data to be read from the teletype, the signal TTYSEL is also asserted. The PTS first copies 8 bits of data to the eight reader data lines (RDR+1 to RDR+128) and then asserts the ACK signal for 2uS. The 920M is required to remove the RDRREQ and TTYSEL signals during this time. The PTS then waits for the paper tape reader / teletype to advance to the next character and thereafter awaits the next request.**

**To make a punch request, the 920M asserts the signal PUNREQ and awaits a corresponding ACK signal from the PTS. If the request is for data to be punched on the teletype, the signal TTYSEL is also asserted. The PTS first copies 8 bits of data from the eight punch data lines and then asserts ACK for about 4uS. Within this time the 920M is required to remove the PUNREQ and TTYSEL signals.**

**The PTS emulates the time each device takes to physically operate. A request for a subsequent operation that occurs before the device would be ready is held up for the appropriate time.**

**The status LED is used to signal emulation status. When running the LED flashes every 1.0 seconds. When an error condition arises, it flashes every 0.25 seconds. The I/O LED is illuminated during a data transfer between 920M and PTS.**

**Logging reports and diagnostic messages are output to the USB serial channel.**

**Serial Channel Protocol**

**-----------------------**

**The PTS emulator expects to be connected via the USB serial link to a terminal emulator program. It sends messages of the following form:**

**Lmessage Logging message (terminated by newline.**

**R Paper tape character read request: the terminal**

**should return a sequence of characters in   
 reply. The first character is interpreted as a  
 count (in the range 1-255) and should be   
 followed by that number of data characters.   
 (This speeds up reading from a file in the   
 emulator.)**

**S Teletype character read request: the terminal**

**should return a single character in reply.**

**P Paper tape punch output request, followed by**

**A single character to punch.**

**Q Teleprinter output request, followed by a**

**single character to type.**

**The characters NUL (0), DEL (255) and NEWLINE (10) are ignored if transmitted.**

**Hardware**

**--------**

**The external case of the 920M emulator provides three LEDs, one switch and a push button. The red LED indicates power is supplied to the emulator (via a USB cable). The green LED is the status indicator as described above. The yellow LED is the data transfer indicator as described above. The switch is labelled LOG and controls the LOG signal (on for logging enabled, off for logging disabled). The push button is labelled RESET and restarts the emulator in the Pico.**

**Electronics**

**-----------**

**The system consists of a pair of Pico processors on a protoboard accompanied by three TXS0108E 8-bit bi-directional level-shifters. One Pico runs PicoProbe to provide a means to download binary programs into the second Pico which runs the emulation. The PicoProbe also provides a USB serial connection for logging output.**

**As the Picos use 3V3 logic, the GPIO pins are connected to the level-shifters for conversion to and from the TTL 5V level of the 920M.**

**The ACK and II\_AUTO lines are pulled down on start up to prevent false signals when the PTS is inactive or not connected. The NOPOWER and LOG lines are pulled up.**

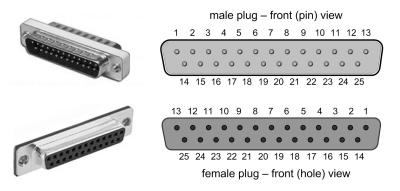
**The pins of the Pico are connected as follows:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Pin** | **GPIO** | **Signal** | **Level Shifter** | **Pin** | **GPIO** | **Signal** | **Level Shifter** |
| **1** | **GPIO0** | **Serial link to PicoProbe** |  | **40** | **VBUS** |  |  |
| **2** | **GPIO1** | **Serial link to PicoProbe** | **Re** | **39** | **VSYS** |  |  |
| **3** | **GND** |  |  | **38** | **GND** |  |  |
| **4** | **GPIO2** | **RDR +1** | **C1** | **37** | **3V3\_EN** |  |  |
| **5** | **GPIO3** | **RDR +2** | **C2** | **36** | **3V3 OUT** |  |  |
| **6** | **GPIO4** | **RDR +4** | **C3** | **35** | **ADC REF** |  |  |
| **7** | **GPIO5** | **RDR +8** | **C4** | **34** | **GPIO28** | **STATUS** |  |
| **8** | **GND** |  |  | **33** | **GND** |  |  |
| **9** | **GPIO6** | **RDR +16** | **C5** | **32** | **GPIO27** | **LOG** |  |
| **10** | **GPIO7** | **RDR +32** | **C6** | **31** | **GPIO26** | **PUNREQ** | **A5** |
| **11** | **GPIO8** | **RDR +64** | **C7** | **30** | **RUN** |  |  |
| **12** | **GPIO9** | **RDR+128** | **C8** | **29** | **GPIO22** | **RDRREQ** | **A4** |
| **13** | **GND** |  |  | **28** | **GND** |  |  |
| **14** | **GPIO10** | **PUN +1** | **B1** | **27** | **GPIO21** | **TTYSEL** | **A3** |
| **15** | **GPIO11** | **PUN +2** | **B2** | **26** | **GPIO20** | **II/AUTO** | **A2** |
| **16** | **GPIO12** | **PUN +4** | **B3** | **25** | **GPIO19** | **ACK** | **A1** |
| **17** | **GPIO13** | **PUN +8** | **B4** | **24** | **GPIO18** | **NOPOWER** |  |
| **18** | **GND** |  |  | **23** | **GND** |  |  |
| **19** | **GPIO14** | **PUN +16** | **B5** | **22** | **GPIO17** | **PUN+128** | **B8** |
| **20** | **GPIO15** | **PUN +32** | **B6** | **21** | **GPIO16** | **PUN +64** | **B7** |

**Pins 1 and 40 are at the USB connector end of the Pico card.**

****

**The external interface is delivered at a female DSub 25 connector, with pins as follows:**

****

**Colour coding of DSub pins:**

**1 ACK Brown 14 RDR +1 Brown**

**2 PUN +1 Brown 15 RDR +2 Red**

**3 PUN +2 Red 16 RDR +4 Orange**

**4 PUN +4 Orange 17 RDR +8 Yellow**

**5 PUN +8 Yellow 18 RDR +16 Green**

**6 PUN +16 Green 19 RDR +32 Blue**

**7 PUN +32 Blue 20 RDR +64 Violet**

**8 PUN +64 Violet 21 RDR +128 Grey**

**9 PUN +128 Grey 22 GND Black**

**10 II/AUTO Red 23 GND Black**

**11 TTYSel Orange 24 GND Black**

**12 RDRReq Yellow 25 GND Black**

**13 PUNReq Green**

**Loopback Test**

**-------------**

**Commented out in the source code for the emulator are a set of routines that provide a loopback test, if a suitable loopback plug is connected. For the 920M the loop back connects the reader data pins to the punch data pins, TTY\_SELECT to II\_AUTO and RDRREQ to ACK.**

**Notes on emulator program**

**=========================**

**The emulator is programmed in C.**

**Two cores are used. One runs the emulation, the second runs a parallel “blinker” routine that flashes the status light according to a global variable blink, which can be set to NO\_BLINK, SLOW\_BLINK or FAST\_BLINK.**

**For some reason probably to do with initialization of the USB system it is necessary to include a short (250ms) wait before performing serial i/o and launching the blinker core.**

**It is necessary to set the GPIO slew rate for the PTS interface output pins to GPIO\_SLEW\_RATE\_SLOW so that the level shifters respond correctly – the Pico has a very fast rise time.**