

Device Bridge

Summary

The original intention of this device is to bridge the old world line printer port world of the Tektronics TDS2024 with the more modern world of USB. This project is loosely based on the work from [Boriz](#) but is a total rewrite of my own design.

Design Goals

- When the print button on the TDS2024 the screen capture/print out should be stored on the bridge
- Captures should be stored locally on the bridge device
- Capture to SD card or local EEPROM
- Allow user to select between SD, EEPROM, and direct transfer
- If SD is not found then only record to EEPROM
- When not printing or after a configured time period show the current time/date on the LCD display
- Reading file transfer header try to detect type and make best guess on proper extension
- Allow user to overwrite the file type
- Allow user to select from name formats from LCD module or Serial over USB
- Allow computer user to list files over serial connection
 - should list SD (if provided) and EEPROM
 - allow transfer of selected files
 - update configuration options
- SD card should support FAT, FAT32 and ExFat.
 - if programming memory allows enable FAT12 as well
- use RTC to maintain current date/time.
 - configurable over serial
 - consider the ability to set from the UI
- consider using one of the secondary serial lines as an RS-232 bridge for SCPI commands

Suggested Improvements

- Enable RTC (DS1307) on the DataLogger shield. When in Idle present the time on the screen
 - <https://github.com/cvmanjoo/RTC>
- Add support for storing files on a W25Q128 (CS pin 3)
 - When SD Card is removed or used selected files may be stored on the eeprom
 - look at using the NASA EEFS.
 - <https://forum.arduino.cc/t/nasa-eeefs/329543/3>
 - <https://github.com/slvajero/EepromFS>
 - https://github.com/feilipu/avr_eeefs
 - <https://www.instructables.com/Introduction-to-LittleFs-Write-LittleFs-Read-Little/>
 - https://github.com/arduino-libraries/Arduino_MKRMEM
- The current read/write process is rather slow
 - look at upgrading to SDFat as well as using a ringbuffer for reading from the parallel port
- consider adding the ability to directly output the content to the serial connection
- use the inputs on the LCD shield to support ejecting/refreshing the SD card
- consider adding prefix profiles that may be added/modified from the serial connection and selected from the interface
- consider the ability to time stamp files
- add the ability to browse/download/delete content from the serial connection
- add proper support for the control lines in relation to SPP IEEE-1284
- add folder management/selection from USB
 - if tied to quick profiles allow selecting from the LCD panel
- look at using the first few bytes to detect the proper file type so they can be named correctly
 - otherwise allow changing the file type from the LCD panel
- add proper error handling

Technical Notes

- Shields
 - LCD and Keypad
 - [OSEPP 16x2 LCD Dispaly and Keyboard Shield](#)
 - Based on suggestion, cut trace for shield pin 10 to prevent possible damage. If you want control there are other options noted in the linked reference.
 - <https://forum.arduino.cc/t/warning-to-users-of-some-vendors-lcd-keypad-shields/94673>
 - Deek Robot Data Logger
 - Parital clone of the Adafruit Data Logger v1 no support for design information from Deek-Robot available.
 - Required extensive rerouting to cleanly support the Arduino Mega 2560
 - Cut traces for shield pins 11/12/13 and reroute SPI lines (MOSI, MISO, SCLK) from ICSP header to 11/12/13 taps respectivly
 - Cut traces for pins 4,5 reouted I2C SCL/SDA to 5/4 respectively (RTC DS1307 support)
 - Added [Windbond W25Q128](#) 16MB SPI EEPROM for local storage
 - Cut static traces for pins 12/13 on the quad buffer (74HC123) from VCC power (pin 14).
 - bodge pin 13 to ground to enable 4th buffer.
 - bodge pin 12 to arduino pin 3 for chip select
 - bodge pin 11 to SPI EEPROM Chip Select
 - bodge MOSI and SCLK from the SD card to SPI EEPROM
 - Custom IO for LPT Port
 - Used 2x13 2.54mm header for 2x13 IDC to DB-25
 - header pins should be mapped as shown in the pinout table
 - note the Strobe line should be mapped to a hardware interrupt pin

Pin Outs

LCD Shield

LCD pin name	Arduino
Reset	pin 8
Enable	pin 9
Data4	pin 4
Data5	pin 5
Data6	pin 6
Data7	pin 7

SD Card Data Logger Shield

Use	Arduino
SPI SCLK	ICSP SCLK
SPI MOSI	ICSP MOSI
SPI MISO	ICSP MISO
SD Chip Select	pin 10
EEPROM Chip Select	pin 3
RTC I2C SCL	I2C SCL

Use	Arduino
RTC I2C SDA	I2C SDA

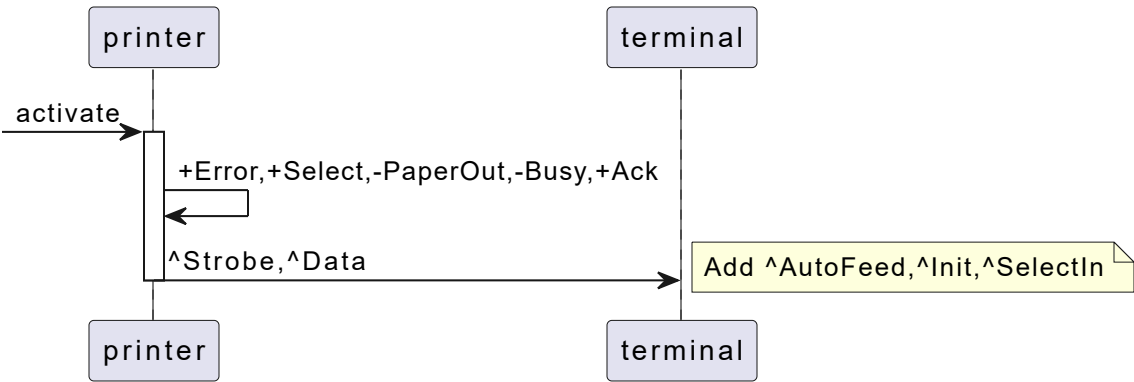
Parallel interface

The [line printer terminal](#) interface is mapped as below

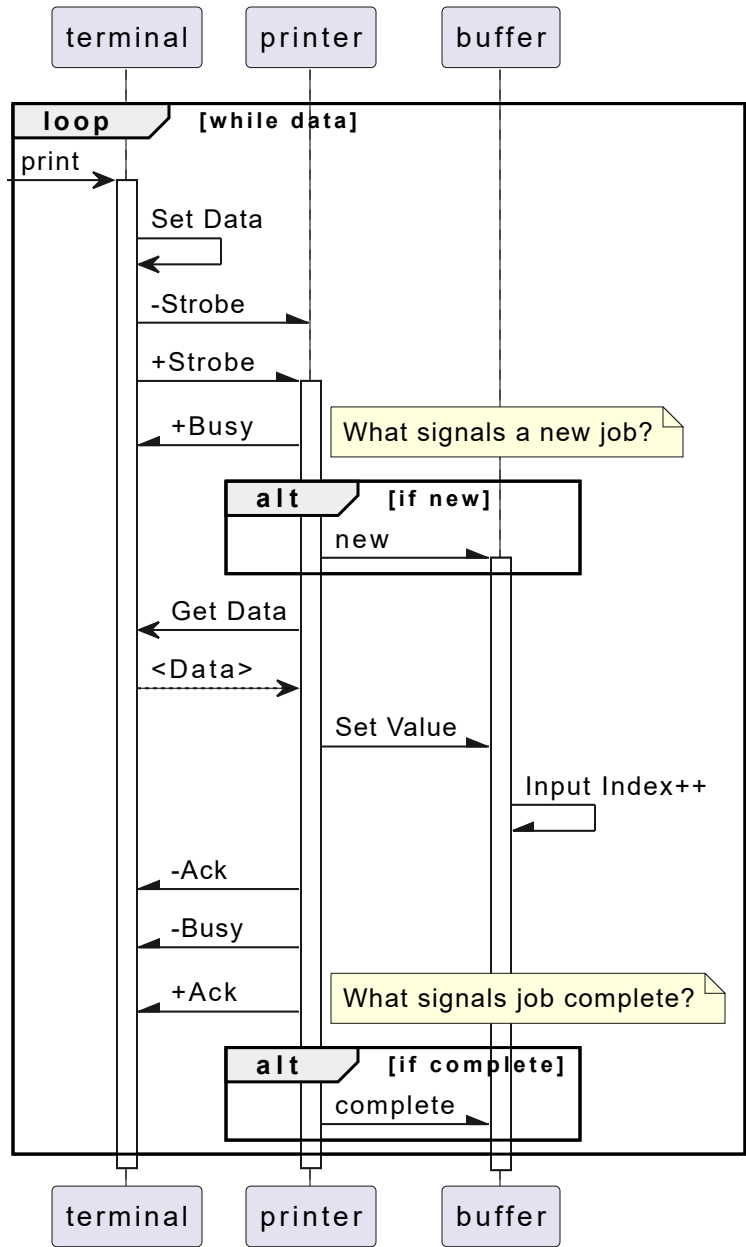
Name	DB25	Arduino	Direction	Notes
/Strobe	1	pin 18	Input	Pullup enabled. Attached to falling edge interrupt.
D0	2	pin 25	Input	Parallel data.
D1	3	pin 27	Input	Parallel data.
D2	4	pin 29	Input	Parallel data.
D3	5	pin 31	Input	Parallel data.
D4	6	pin 33	Input	Parallel data.
D5	7	pin 35	Input	Parallel data.
D6	8	pin 37	Input	Parallel data.
D7	9	pin 39	Input	Parallel data.
/Acknowledge	10	pin 41	Output	Generate falling edge to acknowledge data.
Busy	11	pin 43	Output	Set high on the falling edge of Strobe. Set low after acknowledging data.
Paper Out	12	pin 45	Output	Not used. Forced low.
Select	13	pin 47	Output	Not used. Forced high.
/Auto Feed	14	pin 22	Input	Pullup enabled
/Error	15	pin 24	Output	Not used. Forced high.
/Initialize	16	pin 26	Input	Pullup enabled
/Select In	17	pin 28	Input	Pullup enabled
Ground	18-25	pin 28	Power	

Action Sequence Diagrams

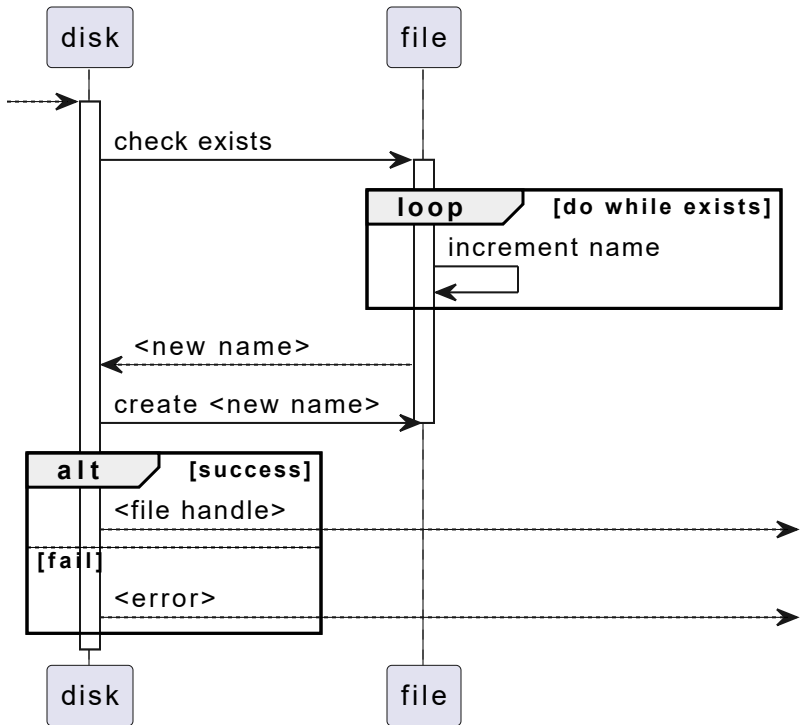
Initialize



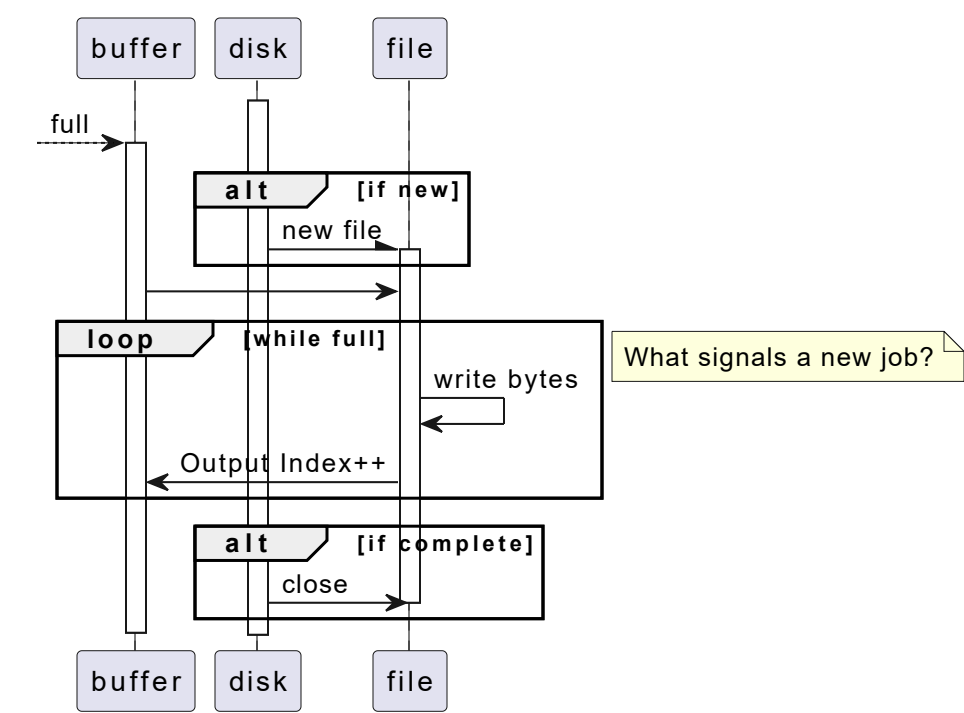
On Print



On New File



Buffer to File



Notes

- [Arduino SD Library](#)
- [SD Card with Logic Level hookup](#)
- [Deek Robot - Data Logger v1](#)
 - <https://www.hackster.io/cowboydaniel/deek-robot-datalogging-shield-tutorial-for-mega-1cd2b1>
 - https://drive.google.com/file/d/1T5GEPMKtLjfiJvdL_DHEbsTk9sAlzThP/view
- [OSEPP - 16×2 LCD Display & Keypad Shield](#)