SEGA Derby Owners Club Card Reader Replacement

September 24th, 2025

Project Overview

YACardEmu is a sanwa card emulator replacement for the popular arcade game, SEGA Derby Owners Club (**DOC**). At its core, is YACardEmu, an open-source GitHub, fan project that emulates a variety of magnetic arcade readers. Designed with extensibility in mind, YACardEmu can be adapted by hobbyists familiar with Windows or Unix-based operating systems.

To support **DOC**, an RS-232 intermediate board is required to connect a SEGA Naomi 1 or 2 system with the Sanwa card reader. In the case of Naomi 2, jumpers must be set so the RS-42x output for communication with RS-232 intermediate board.

Typical setup:

Naomi 1 → RS-232 intermediate board → male DB9-RS232 output -> FTDI USB inserted into Raspberry Pi IOT device or Windows machine.

As of September 24, 2025, YACardEmu supports card insertion and card saving. Caveat, the cards have to be created in another game using YACardEmu and copied into another game to use. Either way, the cards are stored in .bin files and can be renamed, but editing is not a known, known at the writing of this article.

Technical Aim

This document records the process of running YACardEmu on Unix using a **Raspberry Pi 5** with 2GB of RAM in a headless configuration.

Hardware Requirements – refer to Appendix D Hardware and Parts Scenarios.

Community

For ongoing discussion and creative ideas around Derby Owners Club, join us on Facebook. There is also a discord server named Arcade Community.

Contributors: MRDOC and Dexter Bond

Table of Contents

Gettings Started

Hardware and Parts Scenarios:

#1 Solderless

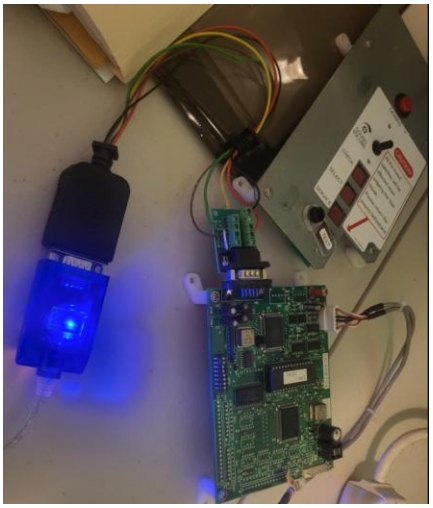
- Purchase two female screw terminals and one rs232 to usb ftdi device.
- This is a solderless solution. If you make a mistake, it is very easy to recover.

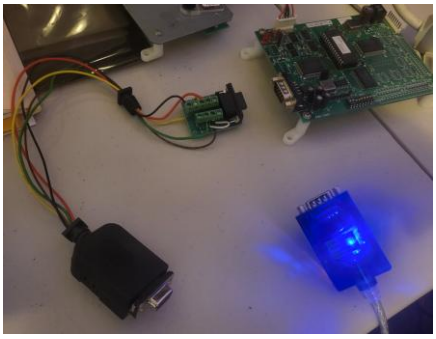
2 female rs232 screw terminals and connectors
1 FTDI compliant device – usb to rs232

USB FTDI RS232 connector image

Raspberry pi 5 with USB FTDI RS232 cable harness

Assembled parts



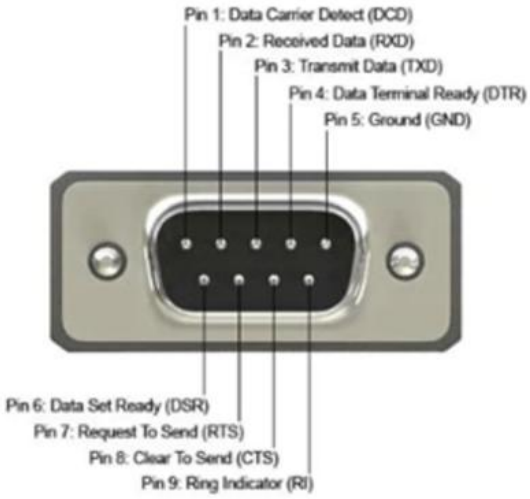


Use the below image and table of pins to wire up the female rs232 screw terminals.

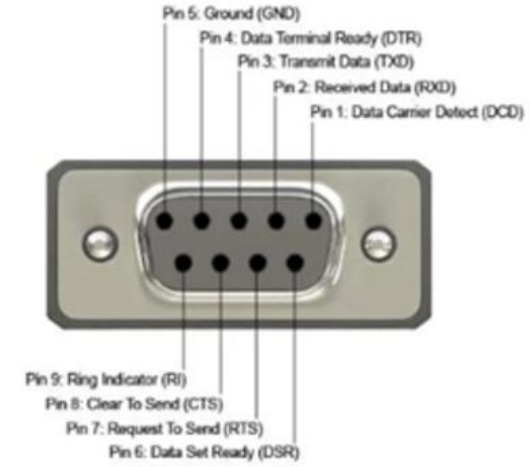
Note: You must jumper the correct pins as instructed below.

RS-232 Source male out from PCB Pin #	RS-232 Signal	Destination Female (Custom DB-9)
1	DCD (Data Carrier Detect)	Pin 1 – Not used
2	RXD (Receive Data)	Pin 3 – TXD
3	TXD (Transmit Data)	Pin 4 – RXD
4	Not used	Pin 4 – Not used
5	GND (Signal Ground)	Pin 5 – GND
6	Not used	Pin 6 – Not used
7	RTS (Request To Send)	Pin 8 – CTS
8	CTS (Clear To Send)	Pin 7 – RTS
9	RI (Ring Indicator)	Pin 9 – Not used
Jumper	CTS & RTS on PCB side	Pins 7,8

RS232 Pinout (Male)



RS232 Pinout (Female)



CABLE PINOUT FROM PCB TO RS232 USB DEVICE

SOURCE FROM/INTO PCB	RS232 USB DEVICE
RXD	TXD
TXD	RXD
GND	GND
RTS	CTS
CTS	RTS

after wiring this up you then want to use a jumper cable between CTS and RTS on the PCB side

#2 Solder solution with connectors.

Purchase **2 db-2 pin female devices and 1 unterminated USB RS232 FTDI device**

1 set of female connectors – requires soldering

1 USB-RS232 device without termination

Steps to install:

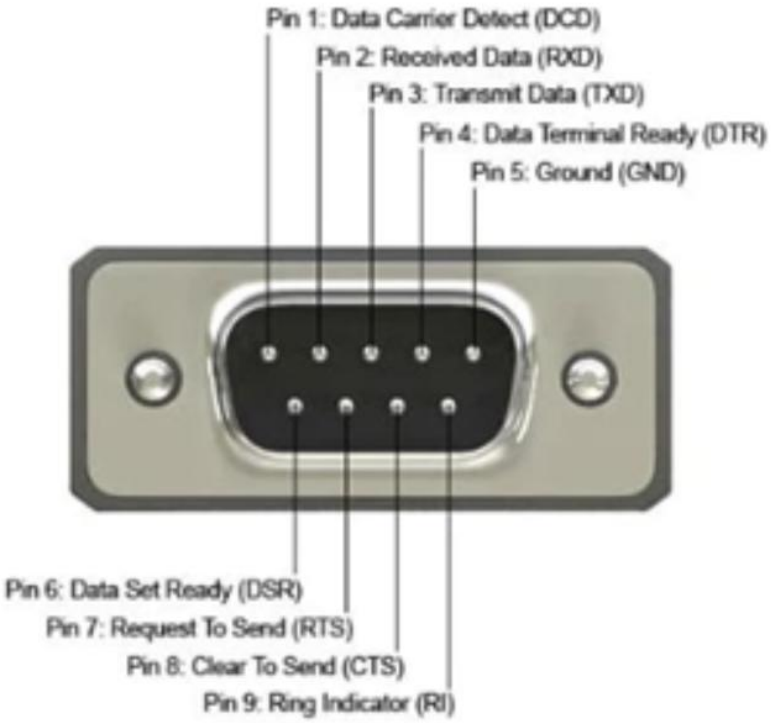
Decipher the below diagrams with pinouts and solder the female connectors with wires and a jumper.

CABLE PINOUT FROM PCB TO RS232 USB DEVICE

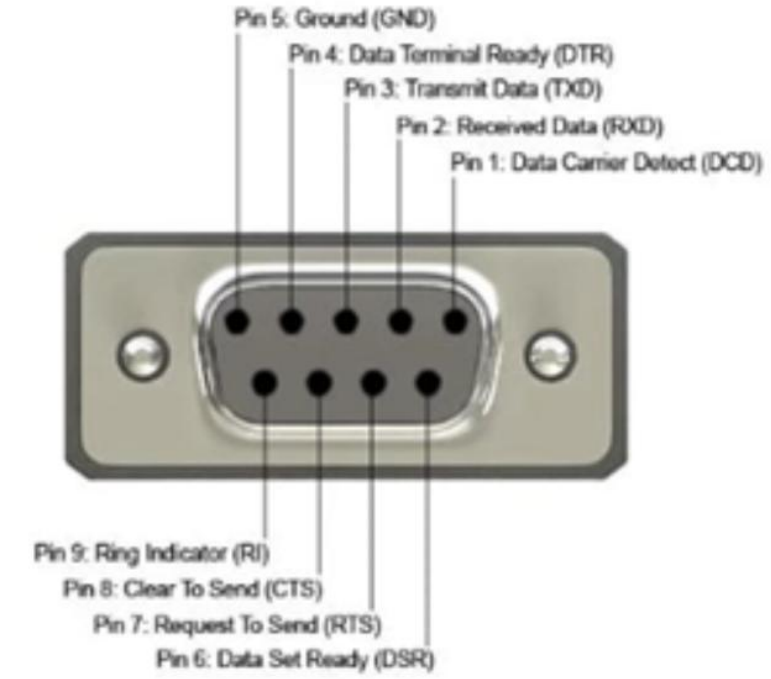
SOURCE FROM/INTO PCB	RS232 USB DEVICE
RXD	TXD
TXD	RXD
GND	GND
RTS	CTS
CTS	RTS

after wiring this up you then want to use a jumper cable between CTS and RTS on the PCB side

RS232 Pinout (Male)

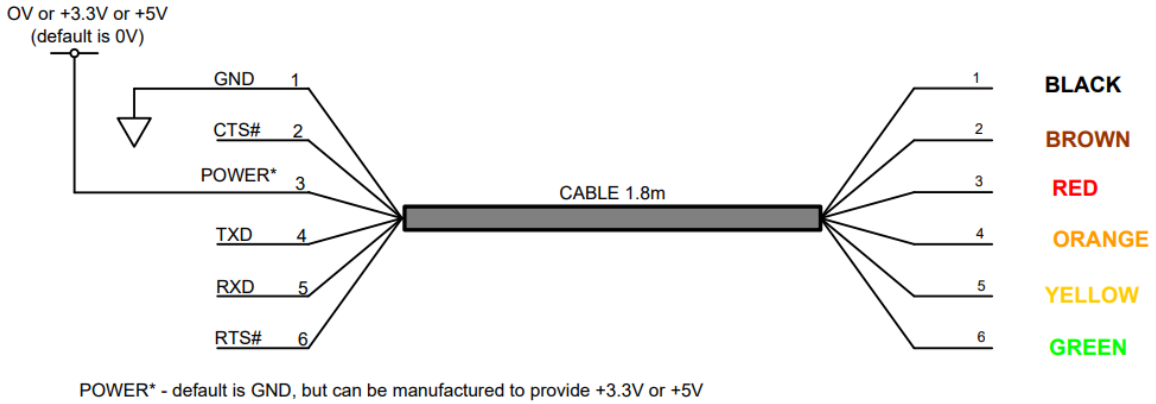


RS232 Pinout (Female)



5.1 USB-RS232-WE-PWR Connections and Mechanical Details

The following Figure 5.1 shows the cable signals and the wire colours for the signals on the USB-RS232-WE cable.



POWER* - default is GND, but can be manufactured to provide +3.3V or +5V

Figure 5.1 USB-RS232-WE Connections

##

Step 1.

#

Getting started

Install latest updates so all commands work.

sudo apt-get update

End of Step 1

Setup dirs.

mkdir myCardReader

cd myCardReader

Create python virtual environment for installation

python3 -m venv --system-site-packages venv

source venv/bin/activate

venv in parenthesis should appear on the left.

if it does not appear, then start over.

Good sample output:

e.g. (venv) piuser@raspberrypi:~

one should install any packages in the python virtual environment

sudo apt install build-essential cmake pkg-config libserialport-dev libsdl2-dev libsdl2-image-dev libsdl2-ttf-dev

github repo yacardemu

git clone --recursive https://github.com/GXTX/YACardEmu

cd YACardEmu

mkdir build

cd build

Build what you downloaded

cmake .. -DCMAKE_BUILD_TYPE=Release

notice the period

cmake --build .

Copy the sample ini file to the correct directory

cp ../config.ini.sample config.ini

Find the usb device inserted into the pi

sudo apt install usbutils

lsusb

Sample output

(venv) piuser@raspberrypi:~/github/YACardEmu/YACardEmu/build $ lsusb

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 003: ID 0403:6001 Future Technology Devices International, Ltd FT232 Serial (UART) IC

Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

(venv) piuser@raspberrypi:~/github/YACardEmu/YACardEmu/build $

Notice in the above, the Future Technology Devices International, Ltd FT232 Serial (UART)

if you are not sure, unplug all usb devices and leave the one you are going to use plugged in.

This is my device, yours may var with the name.

Validate this is the correct device:

ls -l /dev/serial/by-id/

Should list the plugged in serial devices

look for value tty… and in the below sample you will find ttyUSB0

e.g. ttyUSB0, ttyUSB1 and sequentially and so on …

(venv) piuser@raspberrypi:~/github/YACardEmu/YACardEmu/build $ ls -l /dev/serial/by-id/

total 0

lrwxrwxrwx 1 root root 13 Sep 21 12:17 usb-FTDI_FT232R_USB_UART_AB0PJLXJ-if00-port0 -> ../../ttyUSB0

Sample output - notice ttyUSB#. In this case, it is ttyUSB0

(venv) piuser@raspberrypi:~/github/YACardEmu/YACardEmu/build $ dmesg | grep ttyUSB

[4.269508] usb 1-1.4: FTDI USB Serial Device converter now attached to ttyUSB0

(venv) piuser@raspberrypi:~/github/YACardEmu/YACardEmu/build $

In this instance, notice ttyUSB0 is your serial path value for the config.ini file

#---

update the config.ini

Linux

If you have questions, Appendix A contains a minimal, unix configuration.

nano config.ini

change basepath value to where you want to store your horses and images.

This path will also be where you store your horse images in an images folder

Basepath = /home/whateveryourusernameis/cards

serialpath is from above when you ran the command ls -l /dev/serial/by-id

serialpath = /dev/ttyUSB0

Derby owners Club value for card reader is CRP-1231BR-10 - This is the sanwa model number

targetdevice = CRP-1231BR

serialbaud = 9600

serialparity = even

already had something run on 8080 so i changed it to 8081

apiport = 8081

Note: Clara had . which is just current directory. recommended setting is card.bin

autoselected = card.bin

To exit and save your changes in nano

ctrl+o

Answer the file name to write and validate is config.ini and press enter

Exit Nano with Ctrl+x

Create horse images

Sudo mkdir images

images are .png format

e.g. card.bin would have a corresponding card.png and this png would show in the web ui

Setup Kochi fonts in the operating system.

probably the trickiest part as you will need to acquire this file via the internet
kochi-gothic-subst.ttf can be obtained from here.
If the file is not available and you are asked to make a request. Make a request.

mkdir ~\.fonts

cp kochi-gothic-subst.ttf ~\.fonts\ kochi-gothic-subst.ttf

validate YACardEmu exists

command is case sensitive

ls -al YACardEmu

if found, execute the tool with sudo

sudo ./YACardEmu

How To save and insert cards

After starting ./YACardEmu, navigate a web browser to the ip address of the machine you installed with the port number specified in the config.ini

e.g. http://192.168.1.21:8081

Create a horse card:

1. Press yellow start button
2. Breed a horse by selecting sire and dam.
3. Name the foal.
4. Select the Jockey silks and colors.
5. Enter the race and exit.
6. Your card is saved in the directory you specified in the basevalue path of the config.ini file.
    a. Remember, images directory in this folder is what is used to associate your .bin file with your horse and is in .png format.
7. Cards are stored as .bin files after you eject the horse card from the game. Feel free to rename to a new name.bin

Appendix A - Unix minimal config.ini

Note the target device which is the sanwa card reader C1231BR.

```
[config]
basepath = /home/whateveryourusernameis/cards
serialpath = /dev/ttyUSB0 OR w/e your device is
targetdevice = C1231BR
serialbaud = 9600
serialparity = even
apiport = 8080
autoselectedcard = card.bin
```

Appendix B - DB-9 RS232 mapping

CABLE PINOUT FROM PCB TO RS232
USB DEVICE

SOURCE FROM/INTO PCB	RS232 USB DEVICE
RXD	TXD
TXD	RXD
GND	GND
RTS	CTS
CTS	RTS

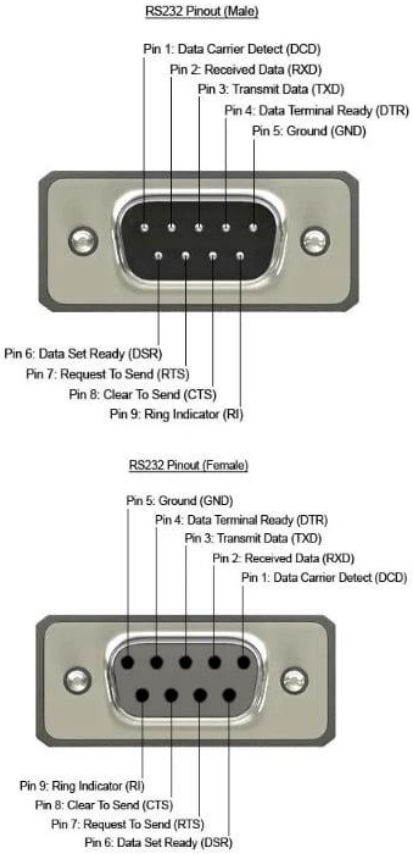
after wiring this up you then want to use a jumper cable between CTS and RTS on the PCB side

Note: PCB side means near the RS232 intermediate board with the male db-9 RS232 interface.

RS-232 wire mapping male out from DB-9 PCB Pin #	RS-232 Signal	Destination (Custom DB-9)
1	DCD (Data Carrier Detect)	Pin 1 – Not used
2	RXD (Receive Data)	Pin 3 – TXD
3	TXD (Transmit Data)	Pin 4 – RXD
4	Not used	Pin 4 – Not used
5	GND (Signal Ground)	Pin 5 – GND
6	Not used	Pin 6 – Not used
7	RTS (Request To Send)	Pin 8 – CTS
8	CTS (Clear To Send)	Pin 7 – RTS
9	RI (Ring Indicator)	Pin 9 – Not used
Jumper	CTS & RTS on PCB side	Pins 7,8

Appendix C - DB-9 standard pins

Db-9 pin male and female standard pin outs



Appendix D - Original RS232 PCB, pinouts and power

For reference, the below tables are the pinouts and wires as shipped from the factory.

RS232 and Power Pinouts for **ORIGINAL** RS232 Conversion board interaction with Naomi1 or Naomi2.

Source (Naomi pinout with Naomi Pin 1 being on the right hand side.)	**Pin**	**Pins on Destination** **Cn7 on pcb board (pins are right to left from edge of board with rs232 male at bottom)**
Naomi1 or 2 RS422 pins	1RX+	6
Naomi1 or 2 RS422 pins	2RX-	5
Naomi1 or 2 RS422 pins	3GND	4
Naomi1 or 2 RS422 pins	4TX+	1
Naomi1 or 2 RS422 pins	5TX-	2
Naomi1 or 2 RS422 pins	6GND	3

Original Naomi Cable to RS232 intermediate board

Source (Naomi pinout with Naomi Pin 1 being on the right hand side.)	**Pin**	**Pins on Destination** **Cn7 on intermediate rs232 pcb board (pins are right to left from edge of board with rs232 male at bottom)**
Naomi1 or Naomi2 RS422 pins	1RX+	6

Naomi1 or Naomi2 RS422 pins	2RX-	5
Naomi1 or Naomi2 RS422 pins	3GND	4
Naomi1 or Naomi2 RS422 pins	4TX+	1
Naomi1 or Naomi2 RS422 pins	5TX-	2
Naomi1 or Naomi2 RS422 pins	6GND	3

Original Power for RS232 intermediate board

PCB board power cable – `cn1`

Pins are left to right from the corner edge of the board.

Source	**Pin**	**Voltage**
CN1	1	5V SEGA Yellow wire
CN1	2	5V SEGA Yellow wire
CN1	3	Blank
CN1	4	Blank
CN1	5	Blank
CN1	6	Blank
CN1	7	GND SEGA White wire
CN1	8	GND SEGA White wire

Troubleshooting

Symptom	Resolution
Prompted to insert a cleaning card?	Pick a digital card from the web interface, click insert and the game will think a physical card has been entered. If asked to turn over the card, insert the same card again.
Please Wait on blue screen	Is your wiring correct?
Other issues	There are parameters built into the system such as -d, -t, -f. For a list of parameters, use ./YACardEmu -h Saving to a file with -f allows for review of the events that transpired and the ability to solve problems. Syntax: ./YACardEmu -f Creates file named **yacardemu.log**
Images do not appear associated with	Create images directory in basepath from config.ini file. E.g. card.bin requires a card.png