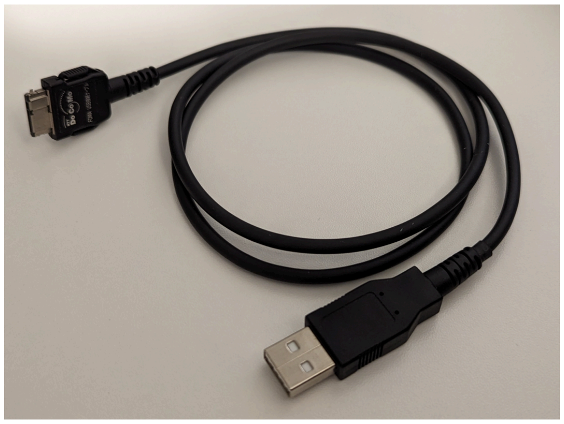
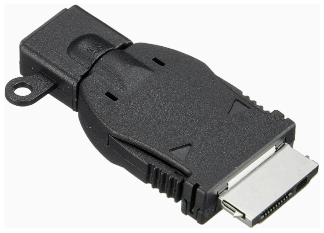
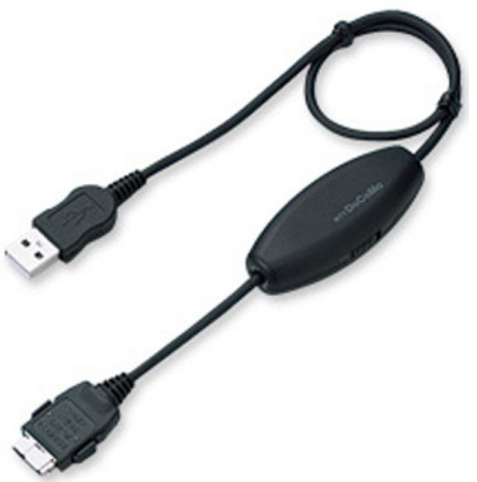
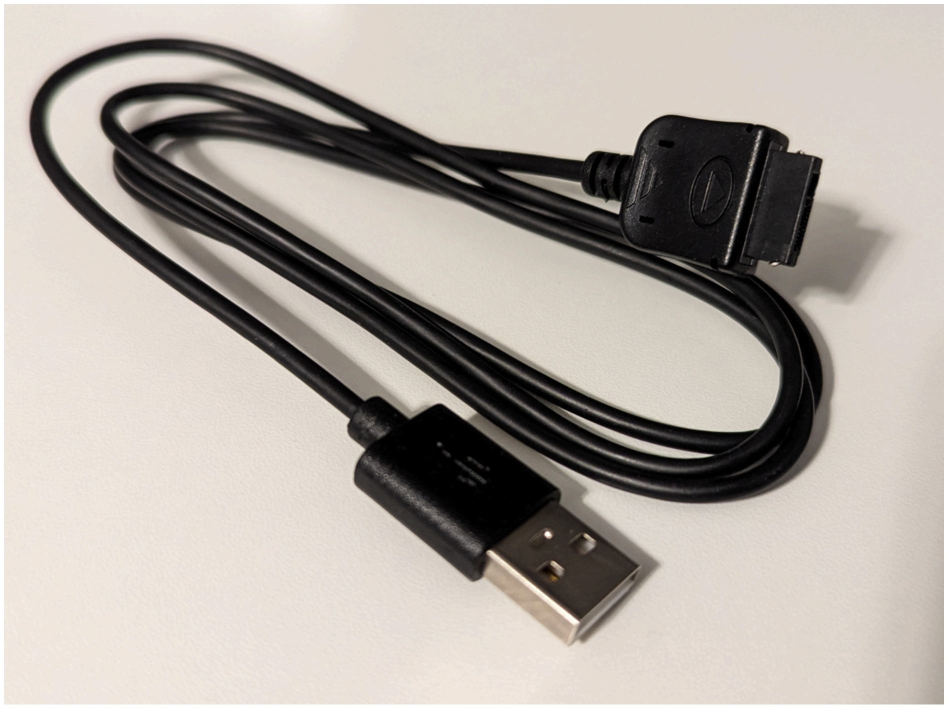
Credits: Xyz, MemoryHunter, NaoyaShinota, RockmanCosmo, LNRC, usernameak, Yuu, TwoSpacesSG, GoodTofuFriday

- **Resources (spreadsheets)**
  - List of preserved i-mode games
  - List of phones that have been dumped with this method
  - *Further game information is on the Kahvibreak Discord server*
- **Making a custom debug cable**
  - You need a compatible FOMA to micro USB cable and soldering iron
    - In the future, you will be able to purchase a custom cable for yourself. That is currently being worked on
    - Official DoCoMo cable
    - ELECOM FOMA to micro USB cable



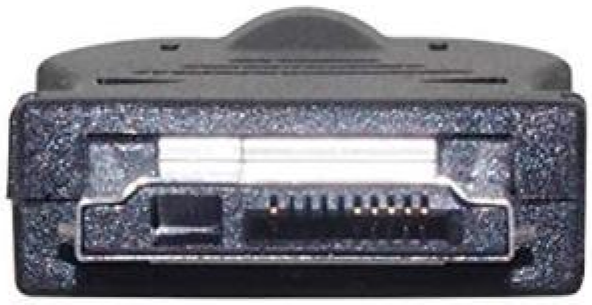
- FOMA USB Connection Cable 02 (power toggle should be in the ON position)

- THESE WILL NOT WORK:
  - 

-

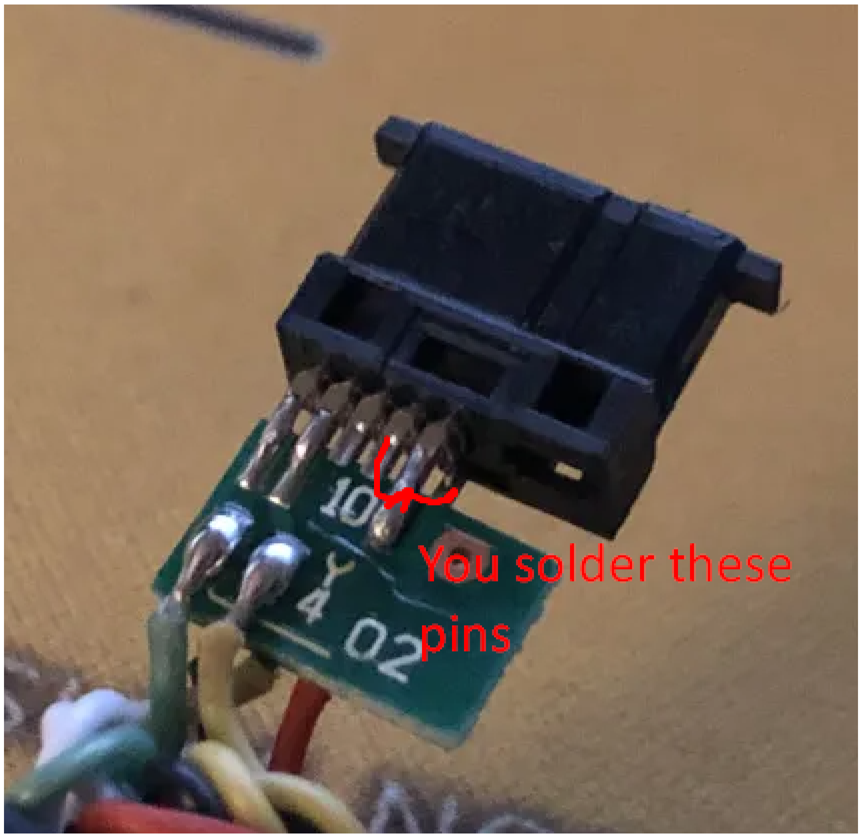
-

- Connect USB power (pin 4) to FOMA power (pin 5), and connect USB ground (pin 1) to FOMA ground (pin 10) as well as pins 8 and 9. Pins 2 and 3 are standalone and are just used for USB data. For the “02 Connection Cable”, you only need to solder 8, 9, and 10 pins.



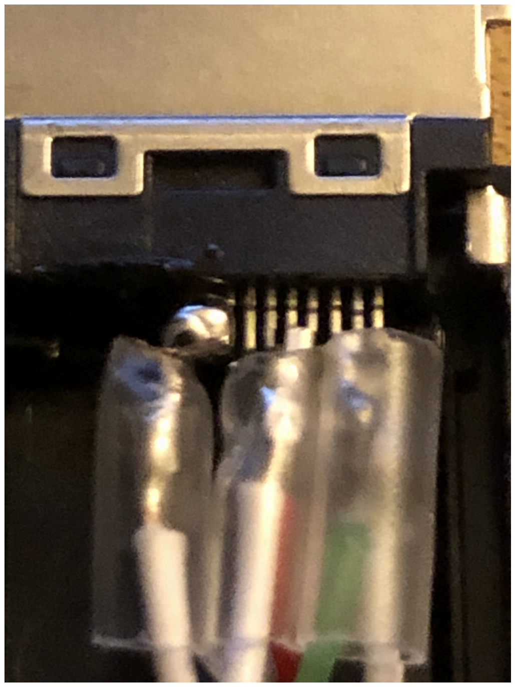
■

- Pin one is the rightmost pin in this picture



■

- As you can see here, only 8, 9 and 10 are soldered for 02 cable.

- 

- **Ripping firmware with xyz's [ktdumper](https://github.com/ktdumper/ktdumper) script**
  - First, install Linux. Most people have used Ubuntu/Xubuntu off a bootable USB
  - Install ktdumper's dependencies as laid out in the GitHub's readme
    - If you run into an error, do this:

      ```
      sudo apt-get install software-properties-common
      sudo apt-get update
      sudo apt-add-repository universe
      sudo apt-get update
      sudo apt-get install python3-pip
      ```
  - Clone the ktdumper GitHub repository
    - Install git

      ```
      sudo apt update
      sudo apt install git
      ```
    - Clone repository

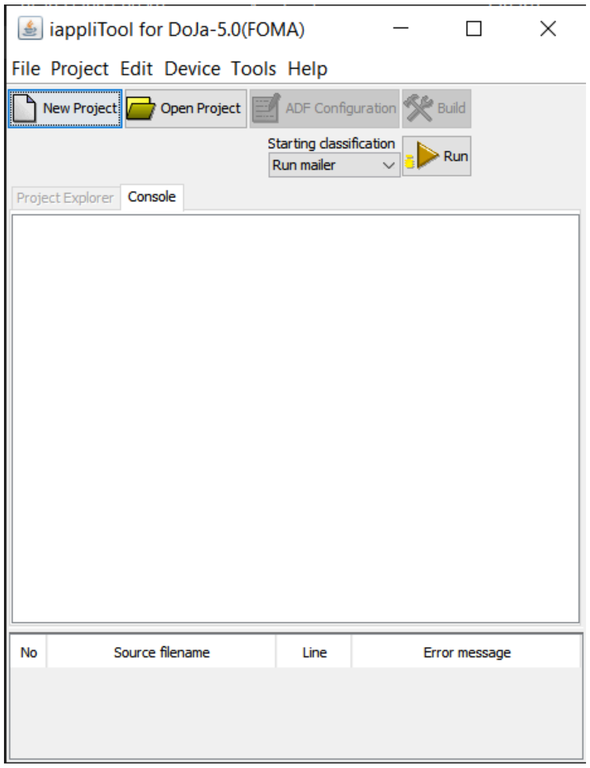
      ```
      git clone https://github.com/ktdumper/ktdumper.git
      ```
  - Plug in the debug cable (connected to your phone without its battery) to your PC

        - Write **lsusb** in the terminal to check if Linux recognizes it. Sometimes the phone shows up as a different phone, but that's not an issue
        - You may need a USB hub if Linux won't recognize the debug cable
    - Dump the phone!
        - See compatible phone models at the top of this document
        - Open the terminal in the ktdumper folder location
        - Write **sudo ./ktdumper.sh** to see the list of supported devices and the necessary payloads
        - Write the commands corresponding to the payloads needed for your phone model
            - Write the device name with the one you're dumping

              ```
              sudo ./ktdumper.sh p321ab dump_nor
              ```

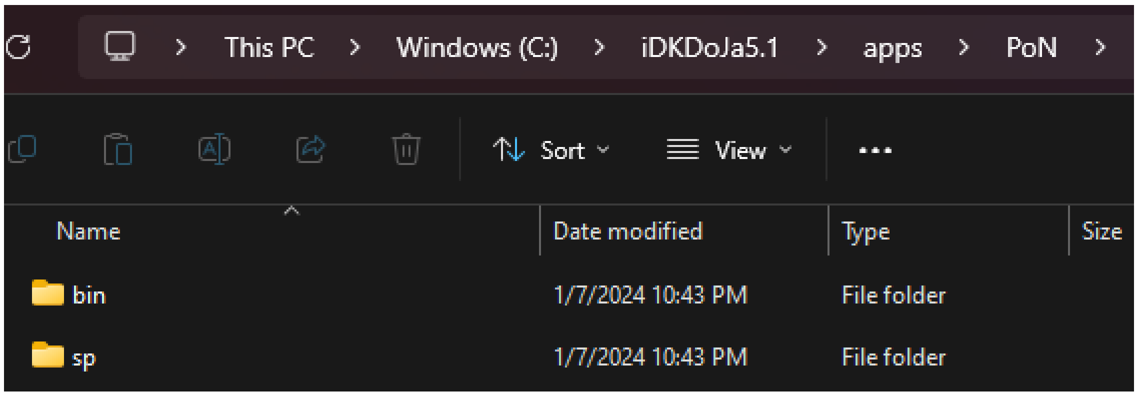
                - In this example, you'd replace "p321ab" with your model name
            - Write the payload that's necessary for your model (**dump_nand**, **dump_nor**, **dump_rom**, etc)
    - Post-dumping procedure
        - Ktdumper creates two files in its directory, an **.oob** and a **.bin** file.
            - If you don't rename or move those files, you could overwrite them if you try dumping another phone
        - Make a separate folder and move those dumped files into it
        - Some raw files may need to be processed to get the filesystem structure to be readable (Some have FTL). Ask in KahviBreak if in doubt.
- **Setting up the DoJa 5.1 SDK's emulator**
    - You need:
        - i-αppli dev tools (specifically the DoJa 5.1 SDK) (for majority of the games, some may need Star 2.0 or DoJa 2.5 International)
            - DoJa 2.5 International can be found here
        - Locale Emulator

        - Oracle JDK 1.8.0, ***32-bit*** Windows build
    - Install the JDK. It should be 32-bit, otherwise, it **will not** work.
    - Install the DoJa 5.1 SDK. Note that the installation path should not have spaces or non-ASCII characters.
    - Try running `bin\doja.exe` in the SDK installation path. If everything got installed correctly, you will see this window:

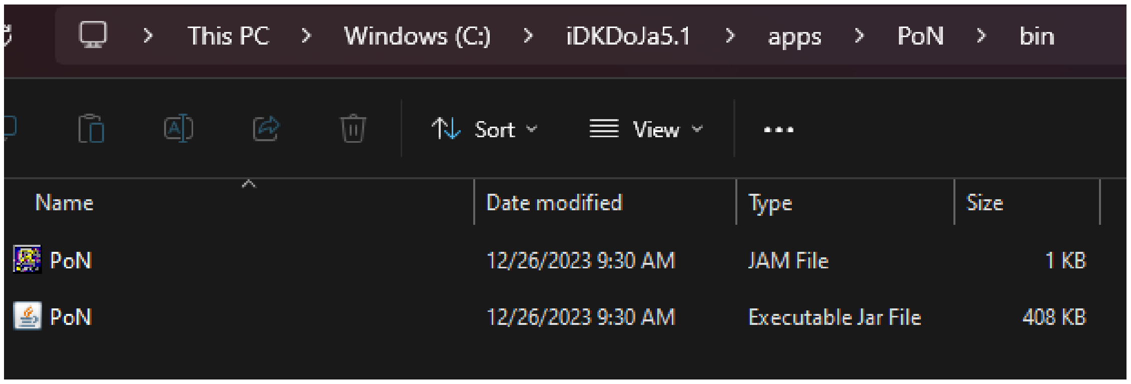


    - Close `doja.exe` for now
    - Install the Locale Emulator. To do that, extract the archive with the utility to an empty folder and run `LEInstaller.exe`.
    - When you need to run the emulator, right-click `doja.exe` and pick *Locale Emulator - Run in Japanese*. This will run with the Shift-JIS character set, which is necessary in order to display text in games properly.
- Installing games to play
    - If you already have a game from somewhere, with a ".jam", ".jar" and an optional ".sp", use DoJaject to easily import the game. Instructions are in the repo. First, you set the DoJa directory (where the SDK is), and then after that, you can select the ".jam"s of games you want to import.

        - After making a folder with the name of the game (**this folder and the game files must have the same name)**, create two folders: bin and sp. Put the jar/jam files in the bin folder, and put the sp file in the sp folder.



Name	Date modified	Type	Size
bin	1/7/2024 10:43 PM	File folder	
sp	1/7/2024 10:43 PM	File folder	



Name	Date modified	Type	Size
PoN	12/26/2023 9:30 AM	JAM File	1 KB
PoN	12/26/2023 9:30 AM	Executable Jar File	408 KB

        -
- If the console tells you **DrawArea** の値が不正です and doesn't boot up the game, that means the game is a "widescreen" game meant for phones with longer screens
    - To fix this,
- Notes for games logging "**SecurityViolation**"
    - Go to Edit -> Trusted Configuration.
    - Codes:
        - 35 - Enable Bluetooth
        - 37 - Enable OpenGL/ES
        - 40 - Enable i-appli call
        - 48 - Enable myMenu request
- Notes for European games
    - For these, you might need DoJa (EU) 2.5 SDK

    - Much of the operation is the same, but scratchpad files have to be **headerless** (without the SP header required for 5.1 SDK). The naming appears to be different: “.scr” extension as opposed to “.sp”, and the scratchpad files have to be numbered starting from 0 (basically, look for what the game tries to create if it does).
    - Locale Emulator also shouldn’t be required for obvious reasons
- Miscellaneous notes
    - If you just dumped a new phone and found folders named “adf” or “jar” or “sp” or files without formats which have contents of the formats mentioned in the previous section, or files with formats “.jar”, “.adf”, “.jam”, “.scr”, “.sp”, then you need to take measures to get them into a working state. Each case is different. Some may require simple renaming, some may require .jam rebuilding (some of the files which are actually “.jam”s are in a weird format which can’t be read without a hex editor). Then, if you do find proper files which are of proper formatting, and you also have an uninitialised (without a 64-byte header) “.sp” file, use spHeader to modify the scratchpad file with a header, which will let the game load without any problems in the SDK.