

# PsNee modchip installation guide

The PsNee modchip is a newer chip that has been under development during the last couple of years, and is still under active development. PsNee modchips should work with every version of the PlayStation 1. They can be made using many AVR processors, including the Arduino Uno, Arduino Pro Mini, ATtiny85, and ATtiny45. This guide covers everything you need to know about choosing, programming, and installing a PsNee modchip into your PlayStation.

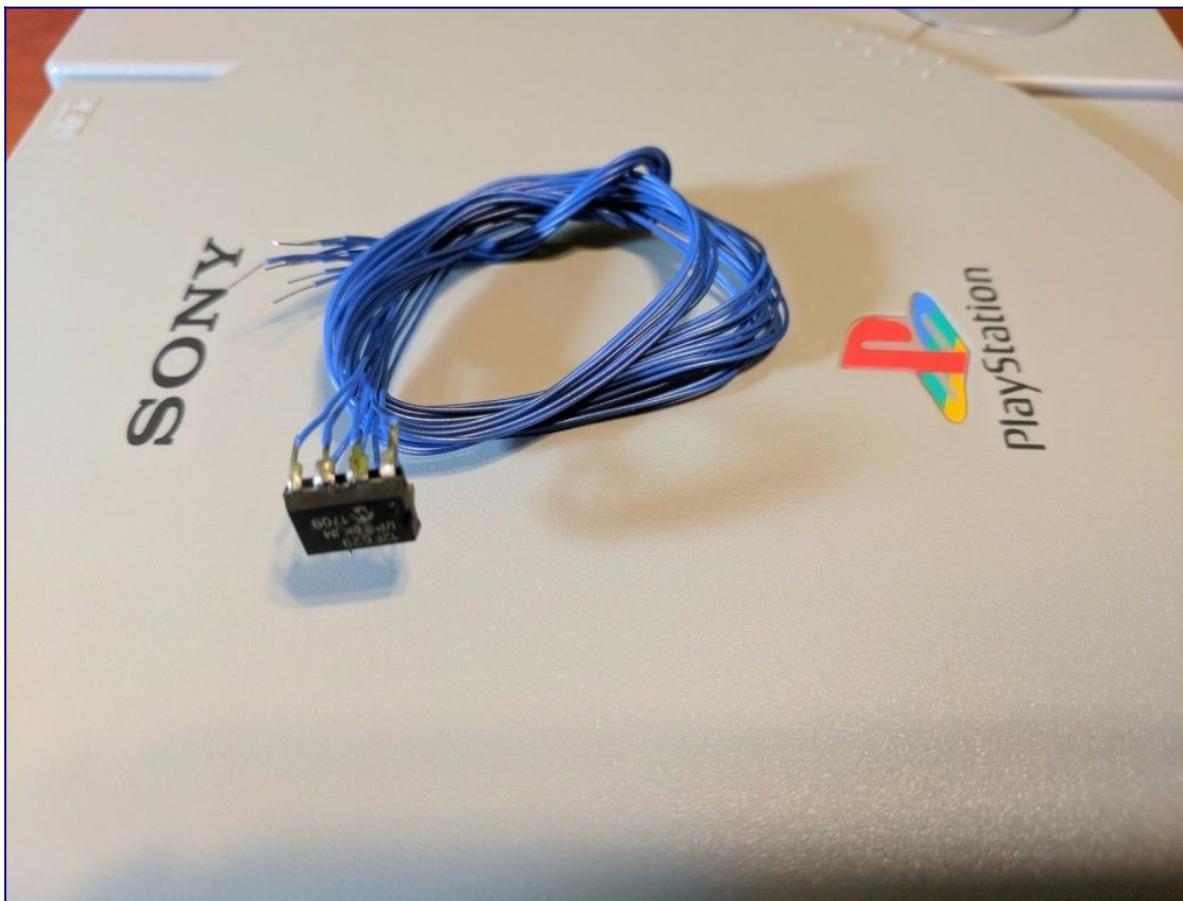
## ***Guide structure***

This guide is structured into several pages. It's organized like this primarily so that comments can be more organized and useful for readers.

This page covers the PsNee modchip. It includes general information about the chip, and then links off to installation guides for each console model.

If you want more information about PlayStation 1 modchips in general [check out this guide](#).

# PsNee modchip installation diagrams



Below is a list of all PlayStation 1 motherboard versions, along with the console model numbers associated with them. You can get a good idea of what board you have by looking at the model number underneath your system (something like SCPH-7501).

To actually know what board you have you'll most likely have to open up your console and look for the board version printed somewhere on the board (something like PU-22 or PM-41).

Once you know what board you have you can click on the board model and you'll be sent to a page with the installation diagram.

- [PU-7](#)
  - All SCPH-1000's
  - Some early SCPH-1001, and SCPH-1002's
  - Some early SCPH-3000's
  - Some early SCPH 3500's
- [PU-8](#)
  - Most SCPH-1001, and SCPH-1002's
  - Most SCPH-3000's
  - Most SCPH-3500's
  - All SCPH-5000's
- [PU-16](#)

- All SCPH-5903's
- PU-18
  - All SCPH-5001's
  - All SCPH-5500, SCPH-5501, SCPH-5502, and SCPH-5503's
  - All SCPH-5552's
  - Some early SCPH-7000, SCPH-7001, SCPH-7002, and SCPH-7003's
  - Some early SCPH-7501's
- PU-20
  - Most SCPH-7000, SCPH-7001, SCPH-7002, and SCPH-7003's
- PU-22
  - All SCPH-7500's
  - Most SCPH-7501's
  - All SCPH-7502, and SCPH-7503's
  - Some early SCPH-9000, SCPH-9001, SCPH-9002, and SCPH-9003's
- PU-23
  - Most SCPH-9000, SCPH-9001, SCPH-9002, and SCPH-9003's
- PM-41
  - Earlier SCPH-100, SCPH-101, SCPH-102, and SCPH-103's
- PM-41 (2)
  - Later SCPH-100, SCPH-101, SCPH-102, and SCPH-103's

## PsNee modchip drawbacks

The biggest problem with PsNee chips is that they aren't very well documented (in terms of things like installation diagrams for every board revision). They also can be hard to find for sale.

This guide, along with my store, will help alleviate both of these issues by providing installation diagrams, and a place to purchase the chips.

## PsNee modchip advantages

The greatest strength of the PsNee modchip is the fact that it's much more modern than PIC based modchips. Using AVR chips means that programming a PsNee chip is much easier, and many people may already have something like an Arduino which can be used as a PsNee modchip.

Using more powerful processors allows the modchip to work with American, European, and Japanese PlayStation 1's with a single chip, no need for a version for each region. It can even patch the SCPH-102's BIOS to allow playing imports (previously this was only available with the OneChip). Note that this BIOS patch only works with ATmega based PsNee chips.

Going forward PsNee modchips will likely become the new standard modchip, especially as the PIC12C508A chip used for more popular chips like the MM3 reaches end of life.

## PsNee modchip source code

If you'd like to program your own PsNee modchips you'll need the source code. It supports the following microcontrollers:

- Arduino Pro Mini (8MHz, or 16MHz) – supported and tested
- Arduino Uno (8MHz, or 16MHz) – supported and tested
- ATtiny85 – supported, untested
- ATtiny45 (8MHz internal oscillator) – supported, tested
- ATtiny25 – supported tested

These chips fall into two categories, ATmega based chips, and ATtiny based chips. ATmega chips cost more, but allow for BIOS patching for SCPH-102 systems, as well as debugging for developers. ATtiny chips cost less, but lack those two features. ATmega chips are also considerably larger in size, making them more difficult to fit into a system when compared to the ATtiny chips.

I'm currently working on developing and testing modifications to the PsNee code to support full BIOS patching functionality with some of the ATtiny chips with more than 8 pins.

The source code is available on the [PsNee Github page](#). It can be compiled very easily using the Arduino IDE.

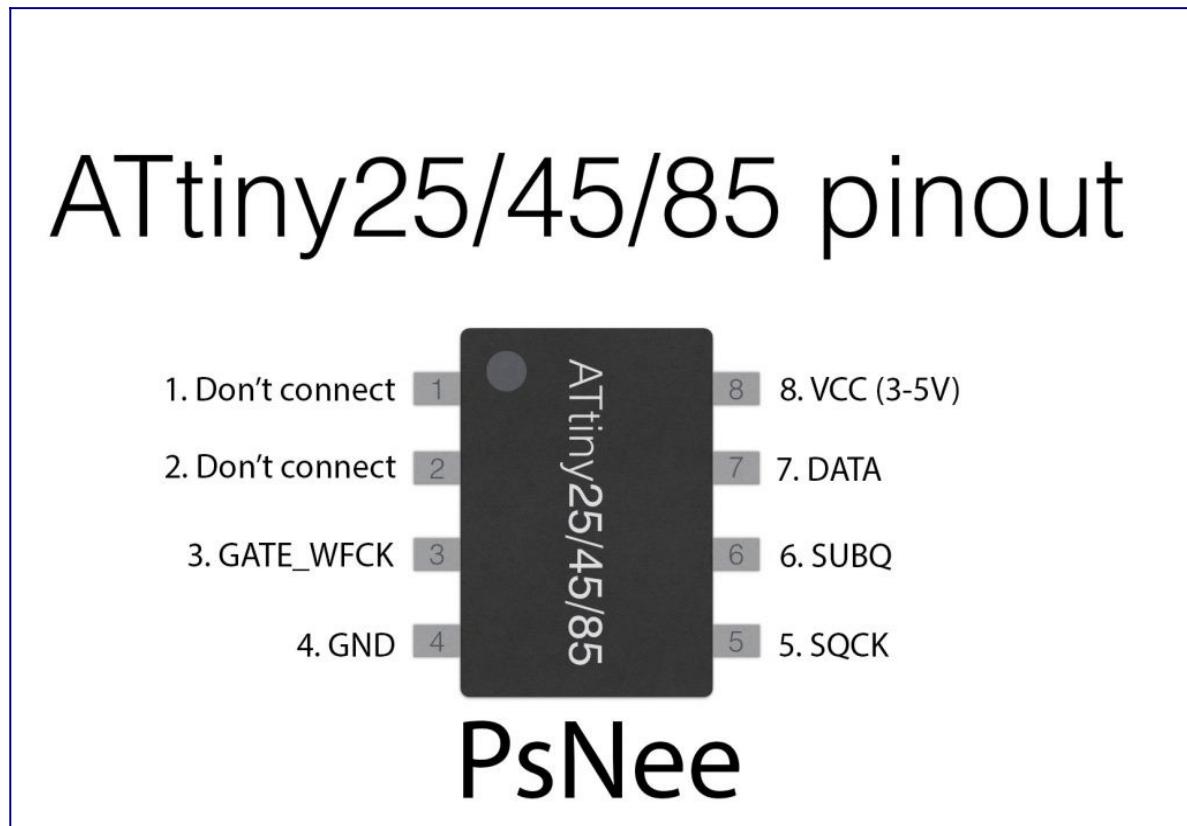
All PsNee chips I sell are preprogrammed.

# PU-7 PsNee modchip installation diagram

The PU-7 was the very first PlayStation 1 board that was released. It is found on all Japanese SCPH-1000's, along with some SCPH-1001's, SCPH-1002's, SCPH-3000's, and SCPH-3500's.

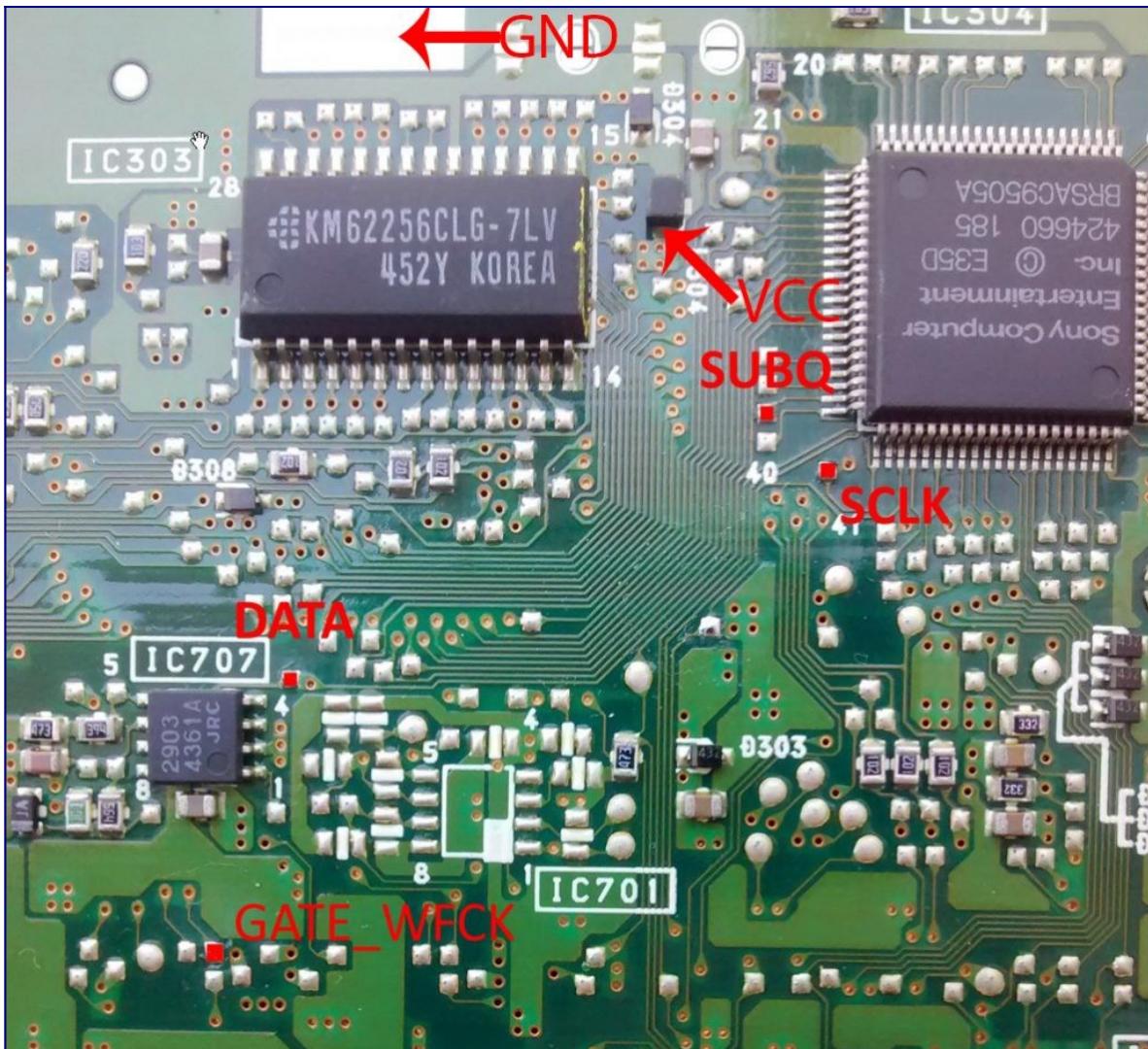
This board is unique because it has the pins for S-Video video output along with the RCA video output ports. The S-Video port was only available on the Japanese SCPH-1000, but the pins are still on other systems with the PU-7 board.

## PU-7 PsNee modchip installation diagram



Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK



Above is the diagram for the PU-7.

All of the points being soldered to are pretty straightforward pads. They are all located on the bottom side of the board. The PsNee is a great chip for the PU-7 because it supports anti-modchip games. The Mayumi V4 isn't compatible with the PU-7 at all, and the MM3 doesn't support stealth mode with the PU-7.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PU-7.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.
- Try to position your chip towards the middle of where all the wires need to go, to minimize wire length.

## Diagram success rate

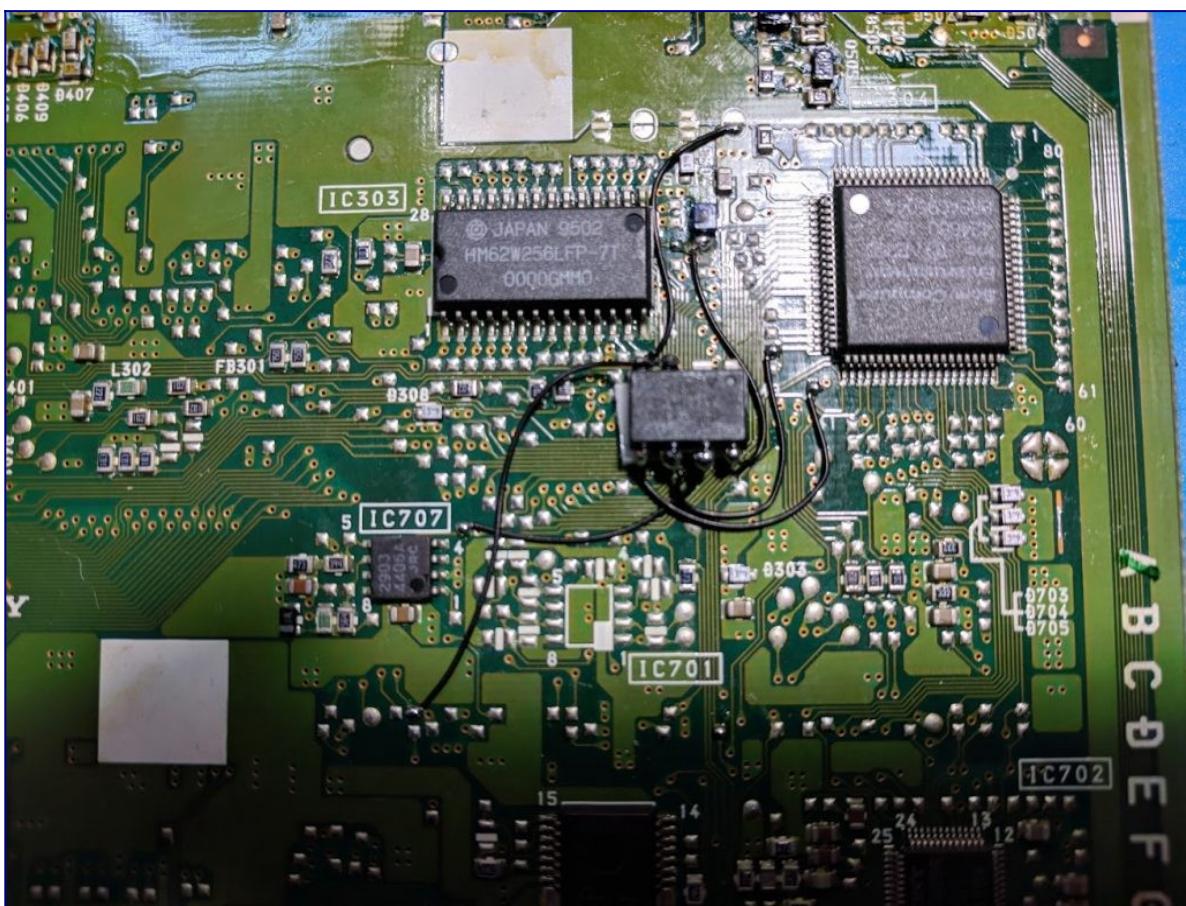
Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. William Quade (me)

## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

This is an installation I did on an SCPH-1000:



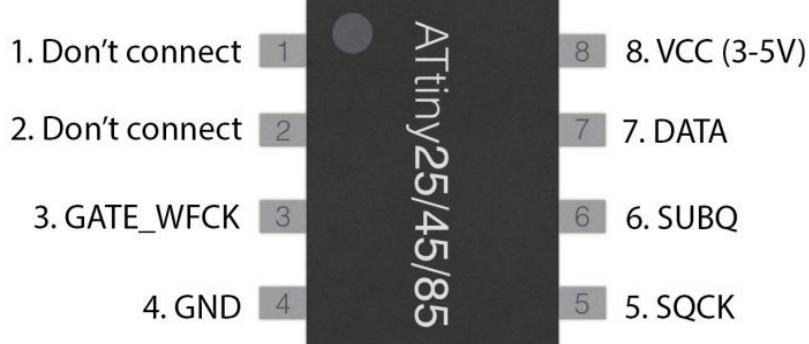
# PU-8 PsNee modchip installation diagram

This was one of the earlier PlayStation 1 board designs that included RCA composite video output ports on the back. Most SCPH-1001's here in the United States had this board inside.

Additionally many Japanese systems had this board including many SCPH-3000, SCPH-3500, and SCPH-5000's.

## PU-8 PsNee modchip installation diagram

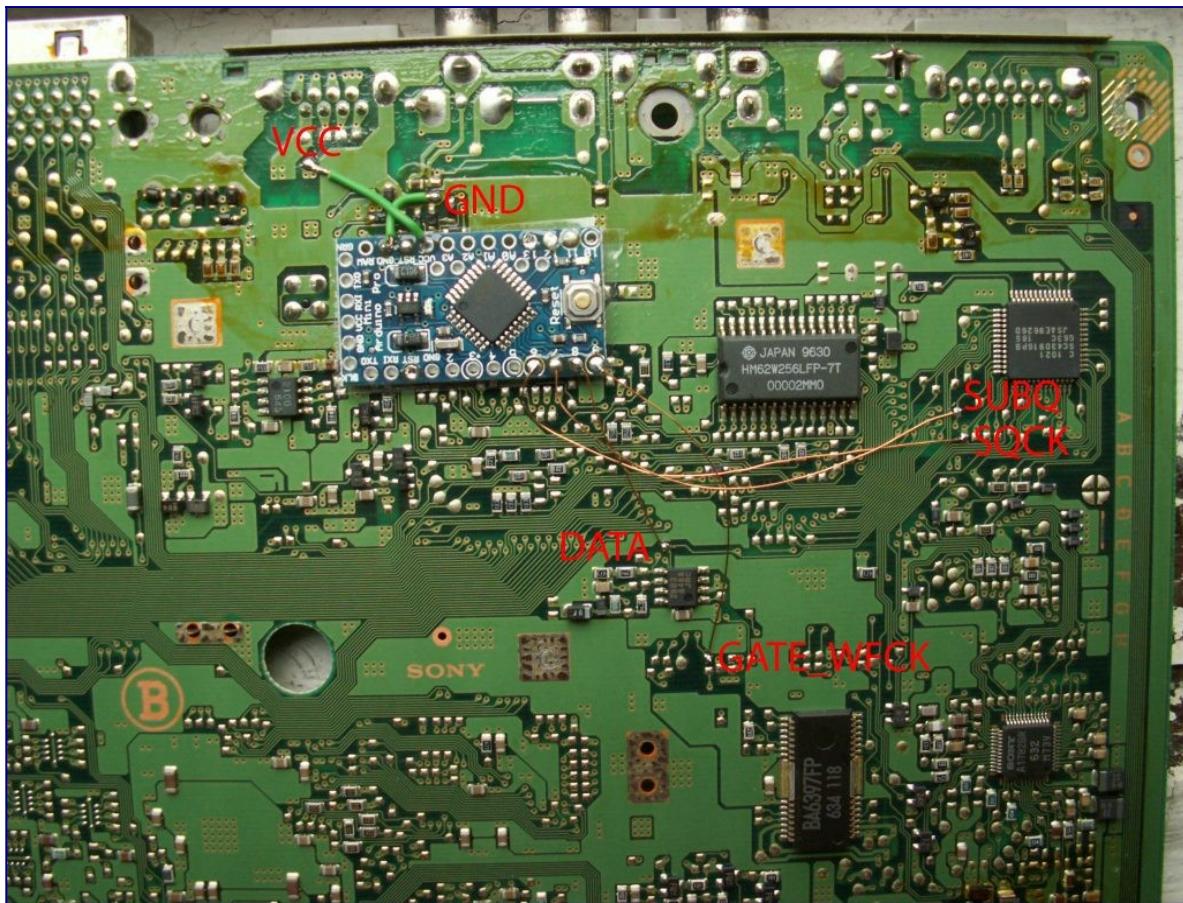
### ATtiny25/45/85 pinout

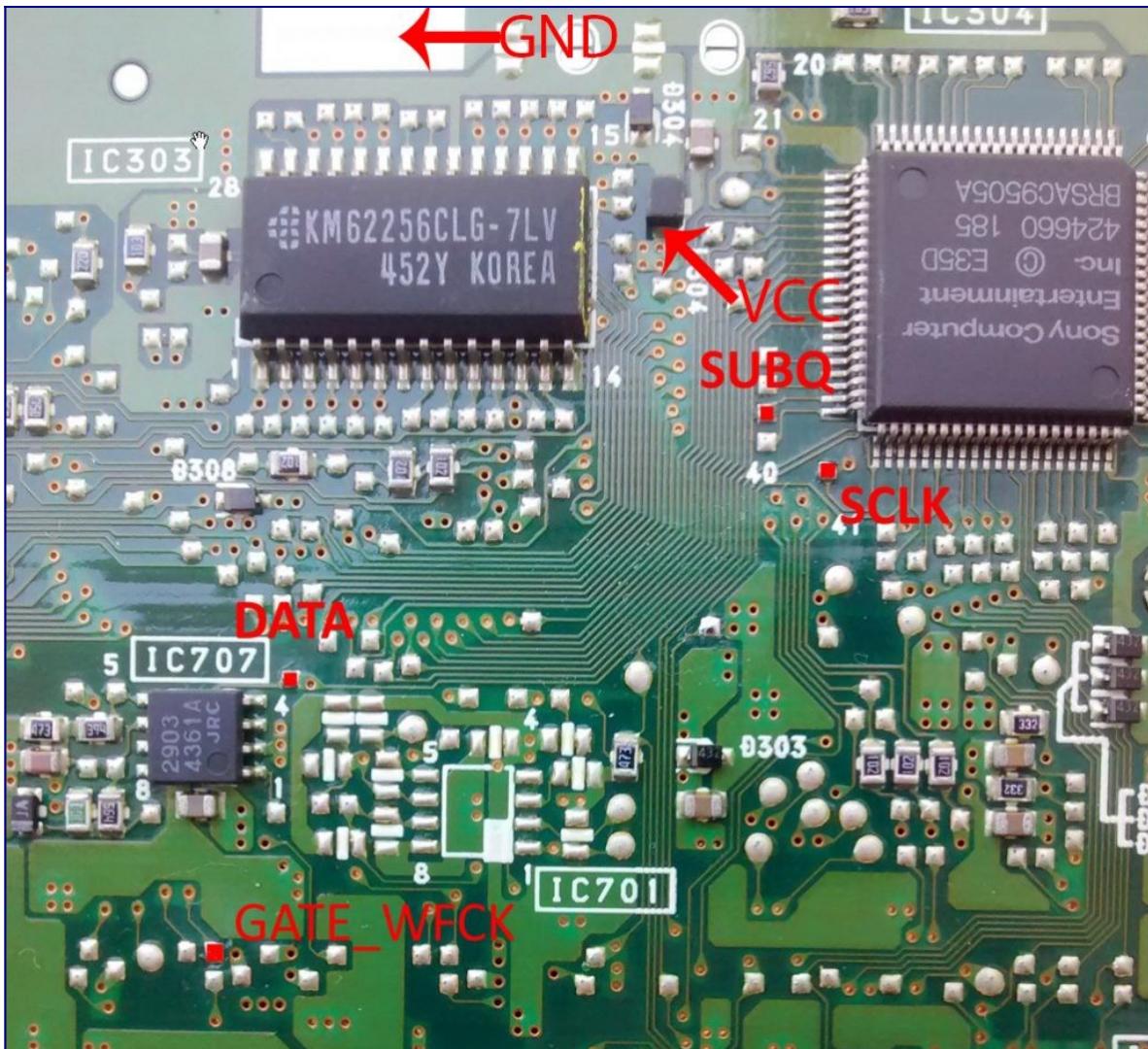


## PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK





Above are two installation diagrams for two different versions of the PU-8. The easiest way to know which you have is to look at the size of the Mechacon controller chip (next to SUBQ and SCLK/SQCK). If it's large, use the bottom diagram, if it's small, use the top diagram.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PU-8.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.
- Try to position your chip towards the middle of where all the wires need to go, to minimize wire length.

## Diagram success rate

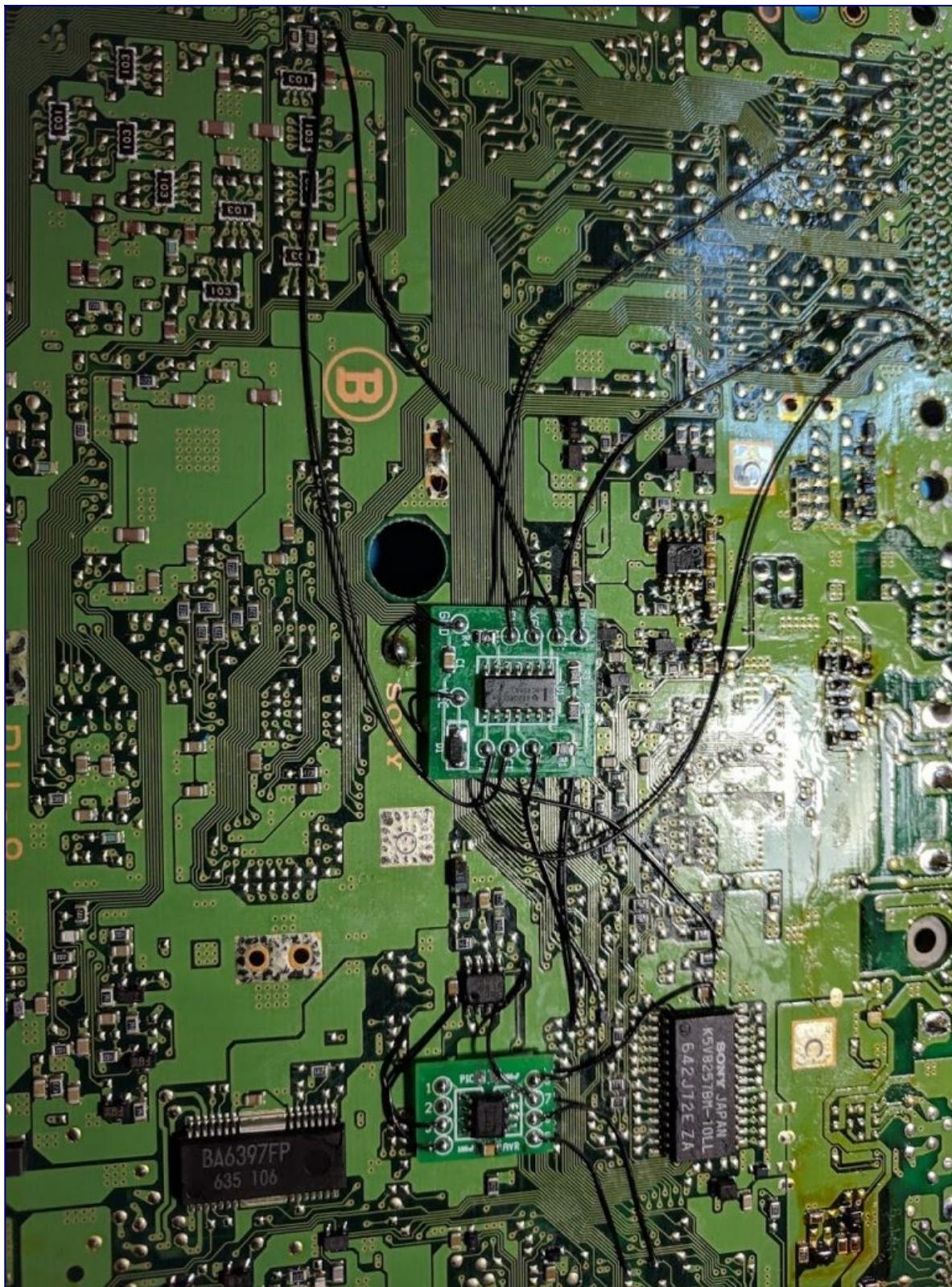
Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. William Quade (me)
2. Nall Wolf

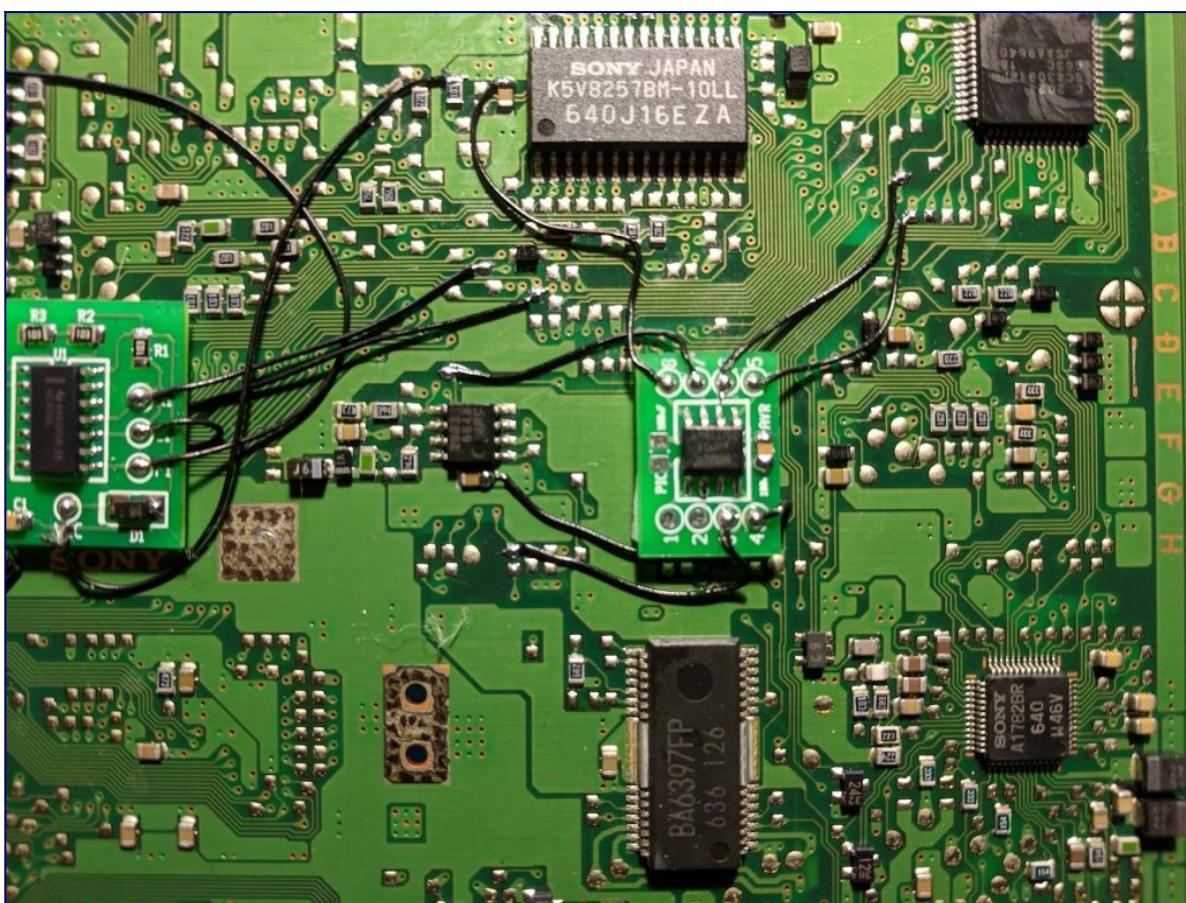
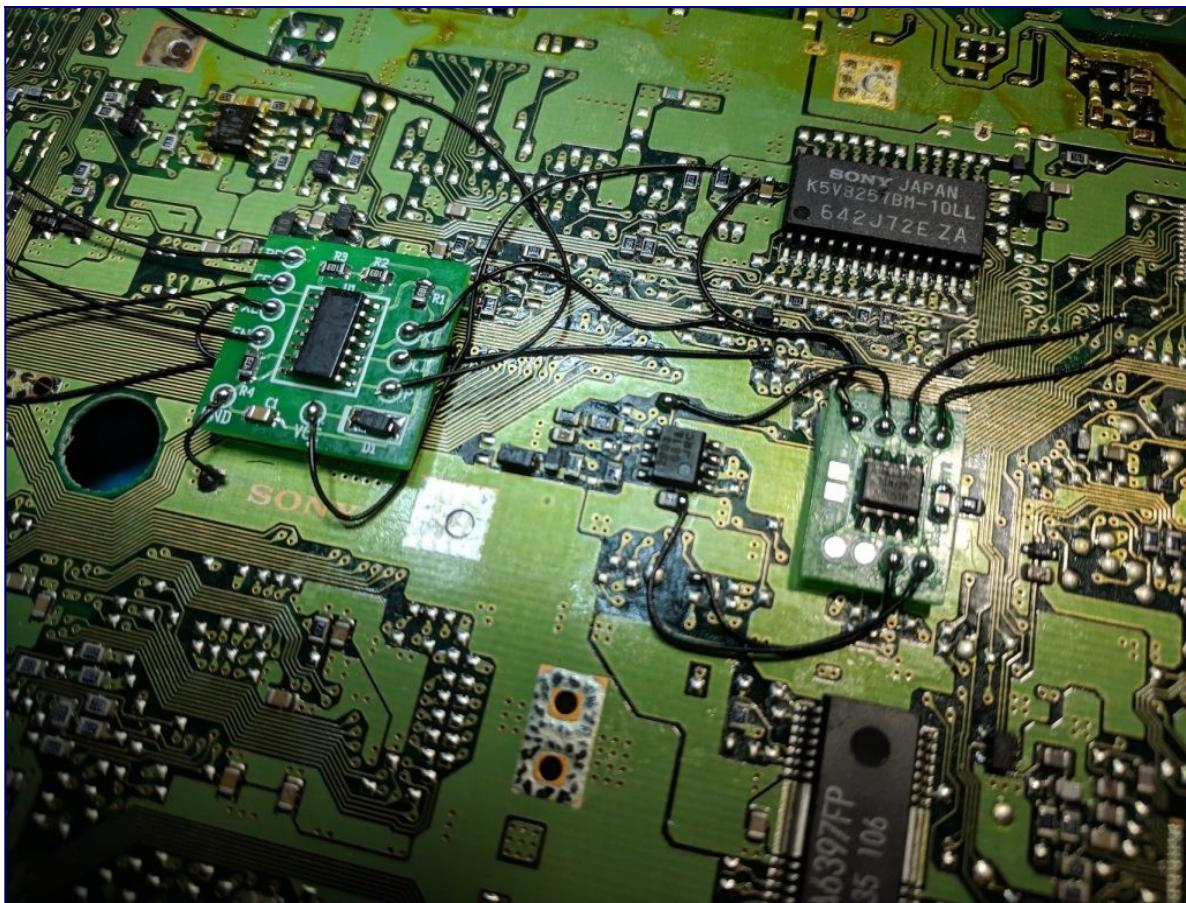
## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

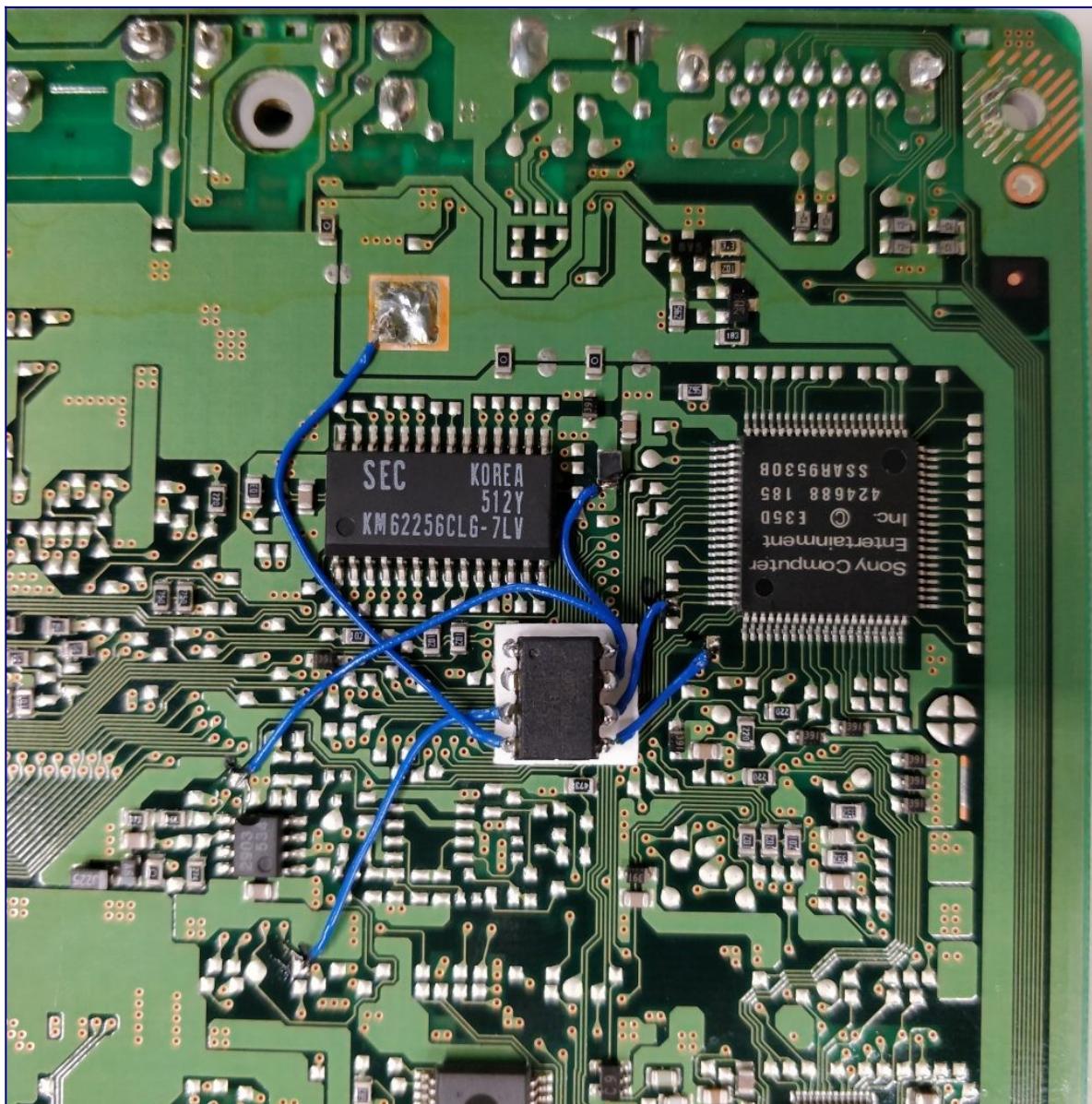
- William Quade (me)



Ignore the larger board in the diagram, it's a PSIO switchboard.



- Nall Wolf



# PU-16 PsNee modchip installation diagram

According to Wikipedia the PU-16 was only found in the SCPH-5903. This is a very unique board in that it was only released for the Asian market, and was only in a single model, and it was colored white instead of grey.

The SCPH-5903 is the only PS1 model that can play video CD movies. The system also has the RCA video plugs like the original SCPH-100x models did.

Apparently there were only around 10,000 of these made, and they were primarily in Hong Kong. So they're fairly rare. Because of this I can't find any PsNee installation diagrams for the system.

Based on what I can find online the board most resembles the SCPH-1000. If you have one of these I'd look at PsNee diagrams for the SCPH-1000 and other earlier boards and try to figure out where things go through trial and error. In other words take a look at the PU-7 and PU-8 diagrams.

**If you have a SCPH-5903 and are interested in having a modchip installed into it I'd be willing to do a free installation so that I can take good pictures of the installation and board (return shipping would be free within the US, but you'd have to pay for return shipping if you're outside of the US).**

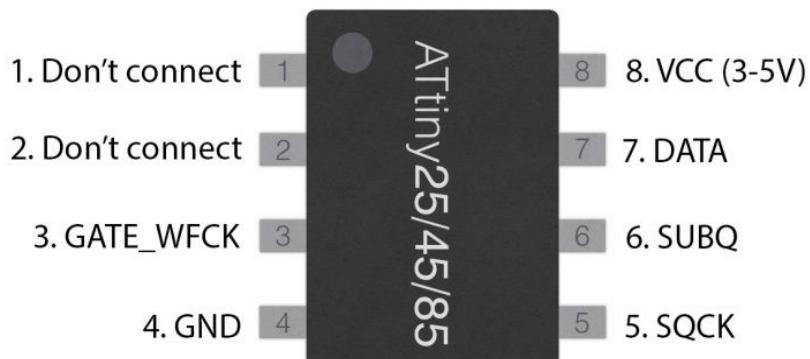
**If you have a SCPH-5903 and are interested in selling it please [contact me](#). I'd be interested in purchasing it if the price is right.**

# PU-18 PsNee modchip installation diagram

This board was used with a wide range of consoles from the SCPH-5000's, and SCPH-5500's, to even some SCPH-7000's and SCPH-7500's.

## PU-18 PsNee modchip installation diagram

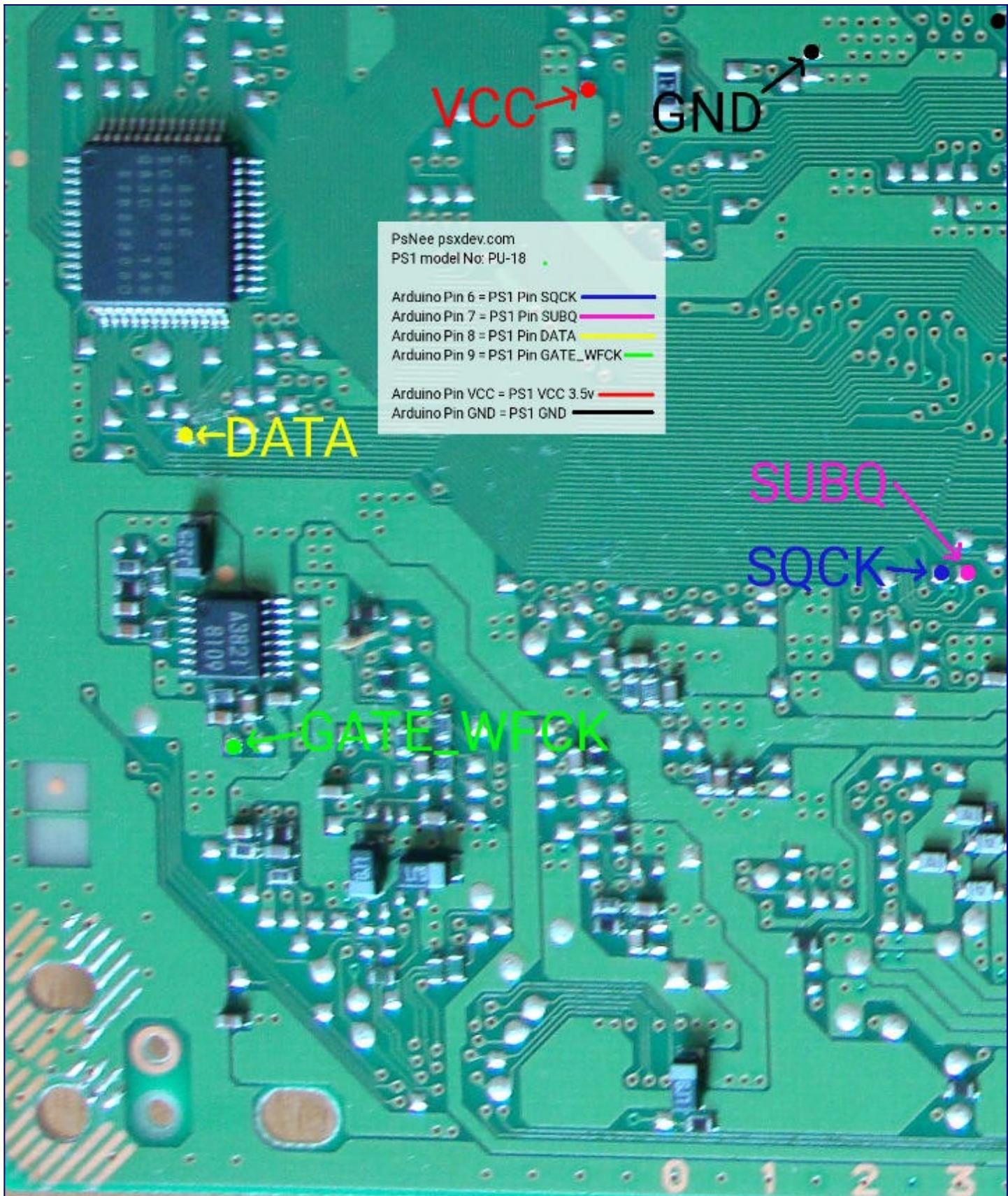
### ATtiny25/45/85 pinout



PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK



Above is the diagram for the PU-18.

All of the points are fairly easy to solder to.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PU-18.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- A good place to put the chip is underneath where the legend is, towards the middle of everything.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.

## Diagram success rate

Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. Nobody yet.

## Example installations

This section displays example installations to help you get a better understanding what a successful installation looks like.

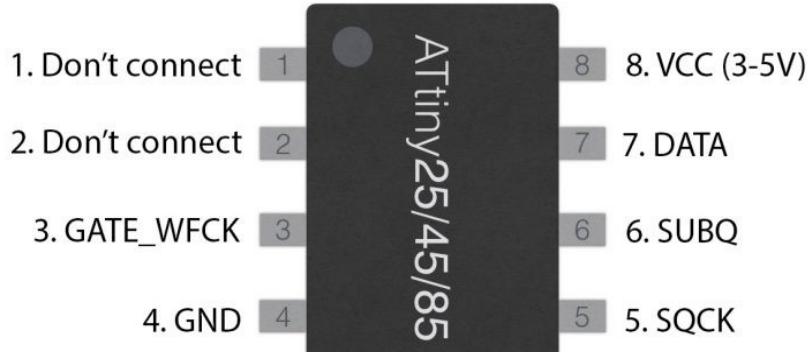
- Nall Wolf

# PU-20 PsNee modchip installation diagram

This board was used exclusively with the SCPH-7000 series of consoles.

## PU-20 PsNee modchip installation diagram

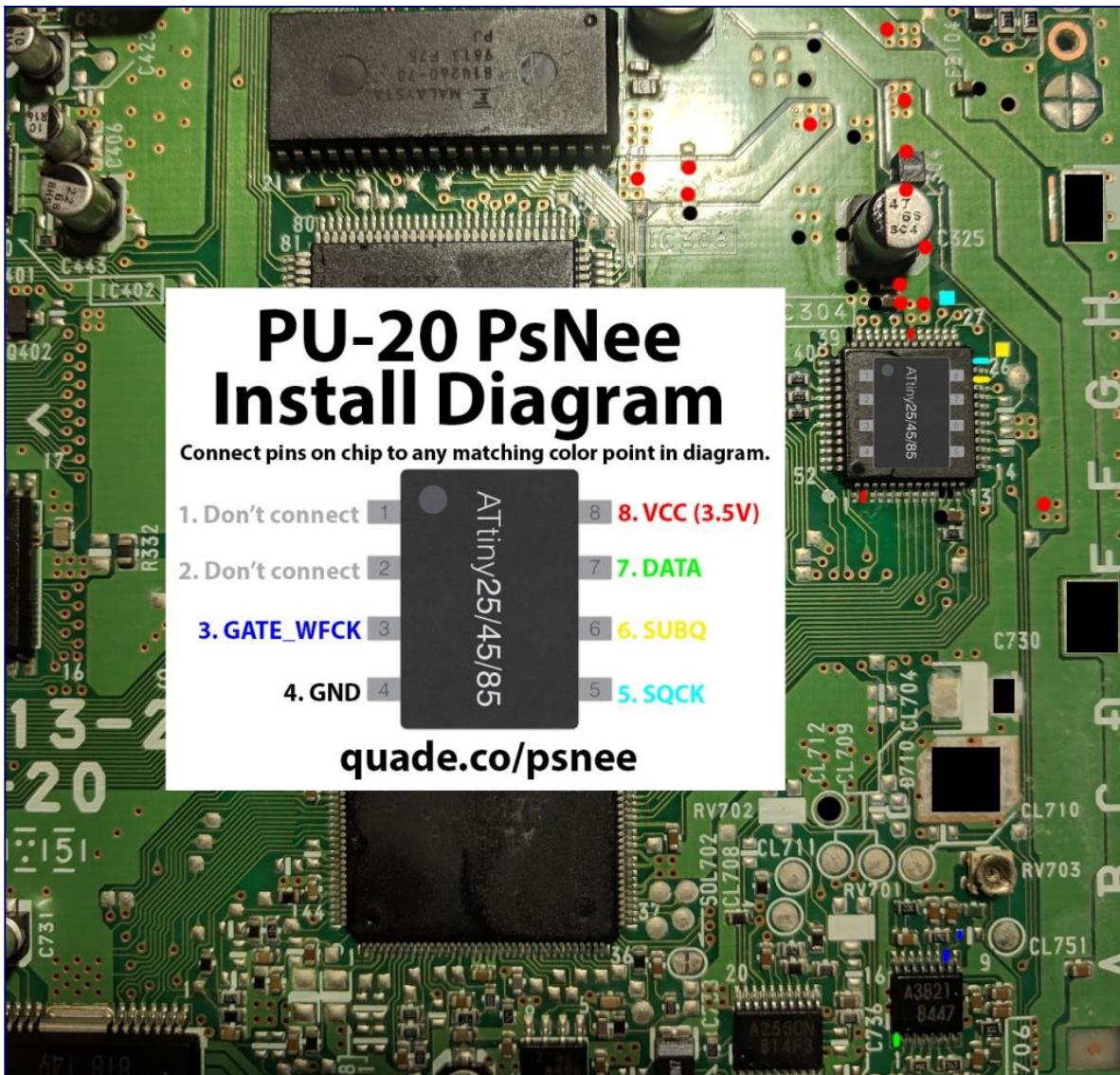
### ATtiny25/45/85 pinout



PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK



Above is the installation diagram for the PU-20. Just match each colored pin label in the diagram with any matching colored point on the board. There is a second image of the chip in the diagram showing a good place to position the chip.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PU-20.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- The DATA pin can be trickier to solder, make sure you don't use too much solder or you risk bridging pins on the chip.

## Diagram success rate

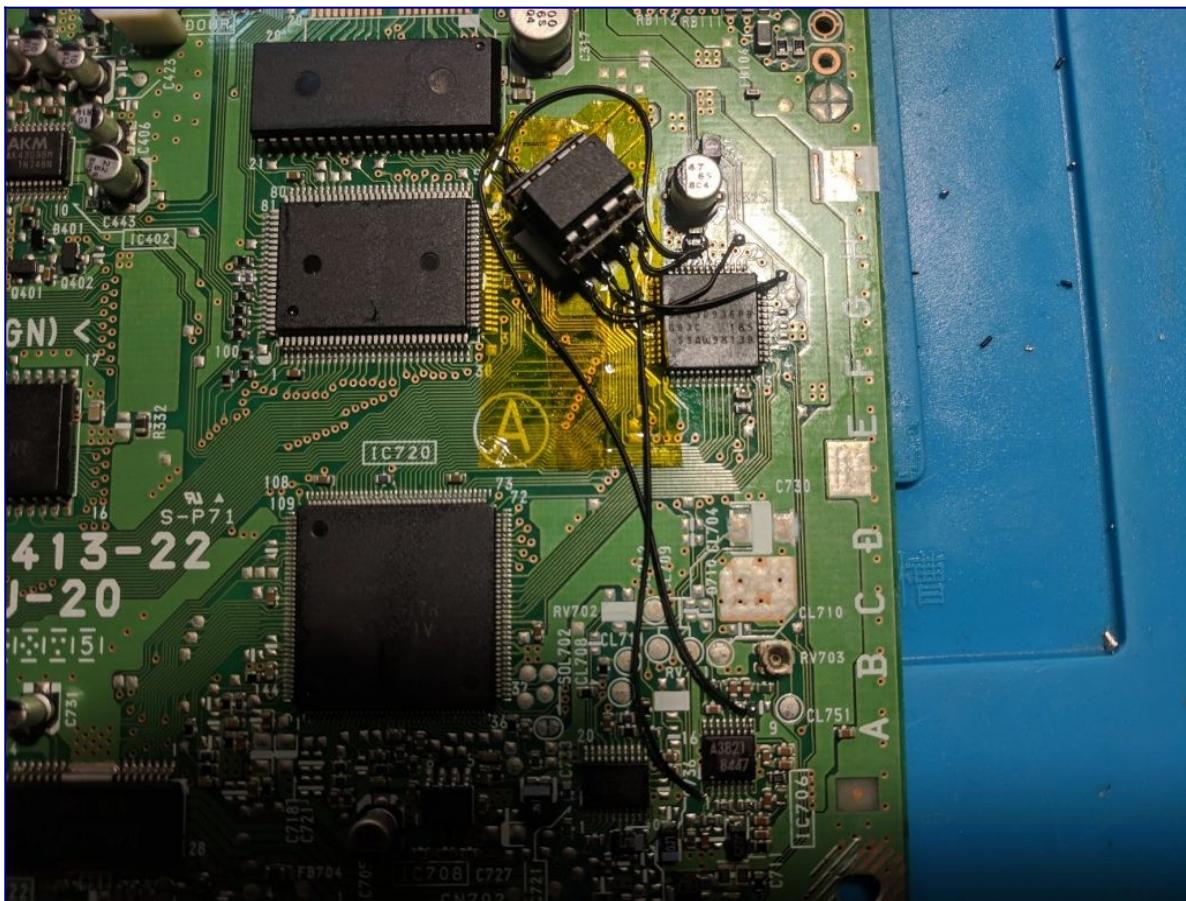
Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. William Quade (me)
2. Retro Gaming Denmark

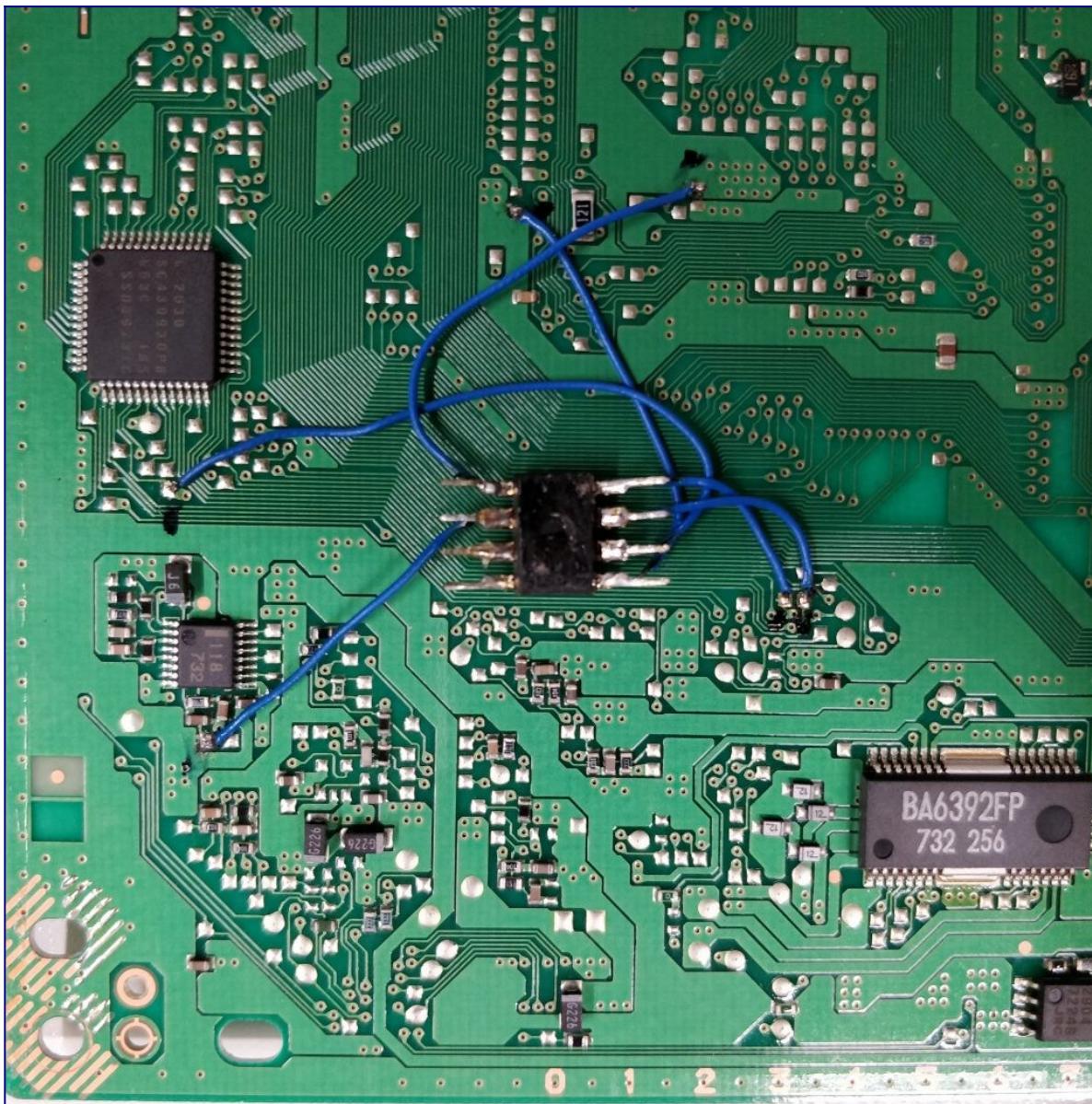
## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

- William Quade (me)



This is an example of an installation I did in a PU-20 using a socket. Using the socket allows me to test chips, and test new PsNee code. Note that the chip is a little too tall and prevents the top metal shield from going on all the way.

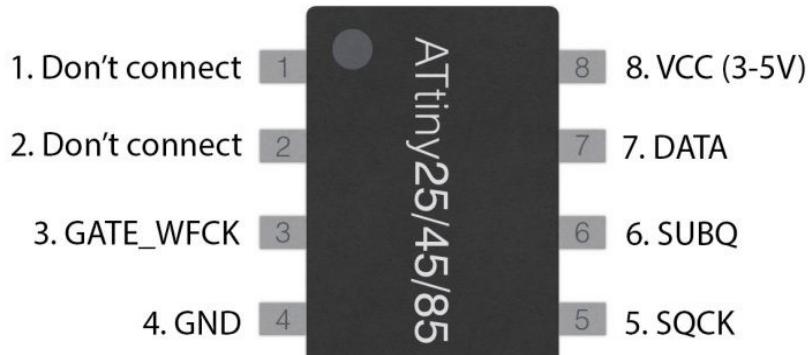


# PU-22 PsNee modchip installation diagram

This is my favorite version of the PlayStation 1. It's in newer systems, so the lasers are more likely to still be in good condition, but the system still has both the serial and parallel ports on the back. This board was used primarily with the SCPH-7500 series of systems.

## PU-22 PsNee modchip installation diagram

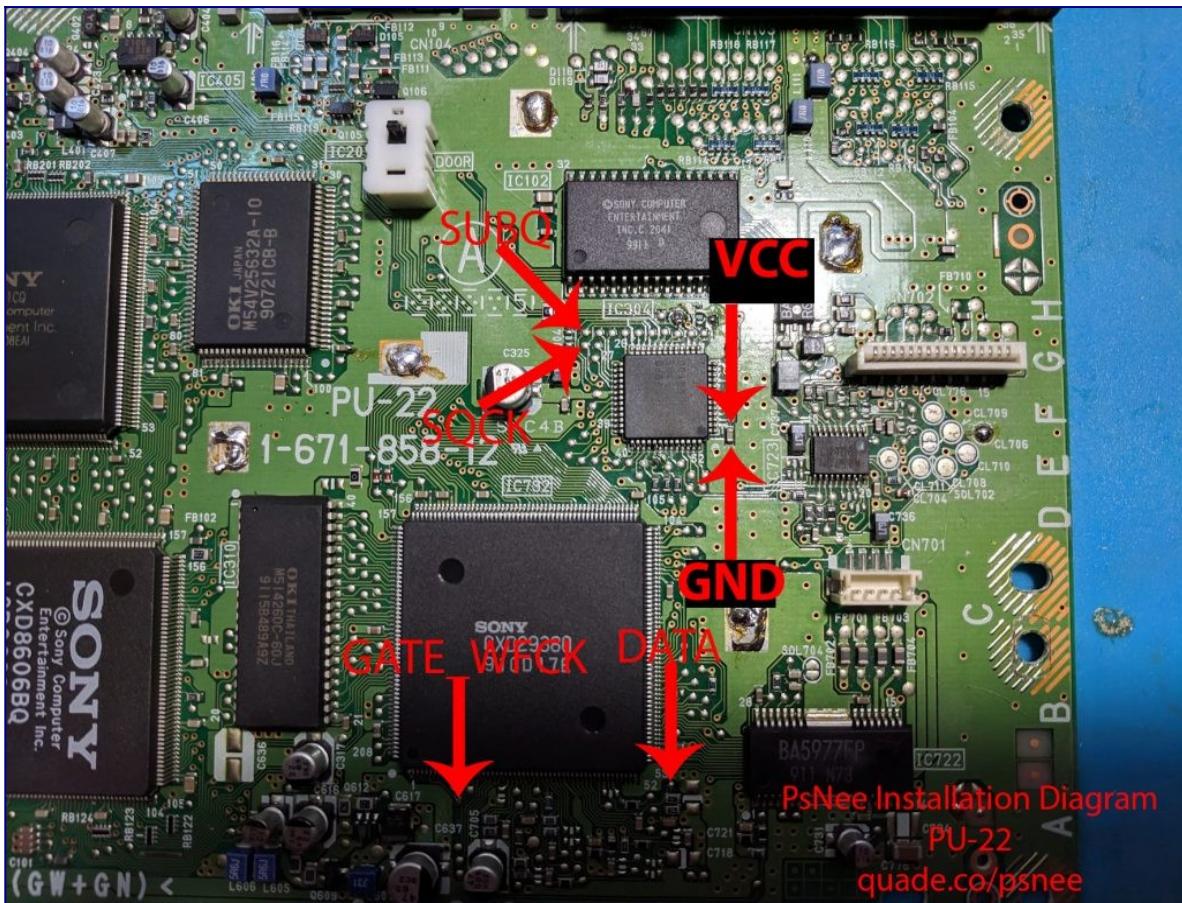
### ATtiny25/45/85 pinout



PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK



Above is the diagram for the PU-22.

This diagram is fairly straightforward. Be careful when soldering to the capacitor that connects to VCC and GND. In particular, the GND point can easily be bridged to nearby pins/pads.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PU-22.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.
- Placing the chip on top of the chip in between SUBQ, SQCK, GND, and VCC is a good spot.
- Watch out for solder splatter when you are desoldering the metal shield on top of the board. It isn't necessary to put it back on, but I always do.

## Diagram success rate

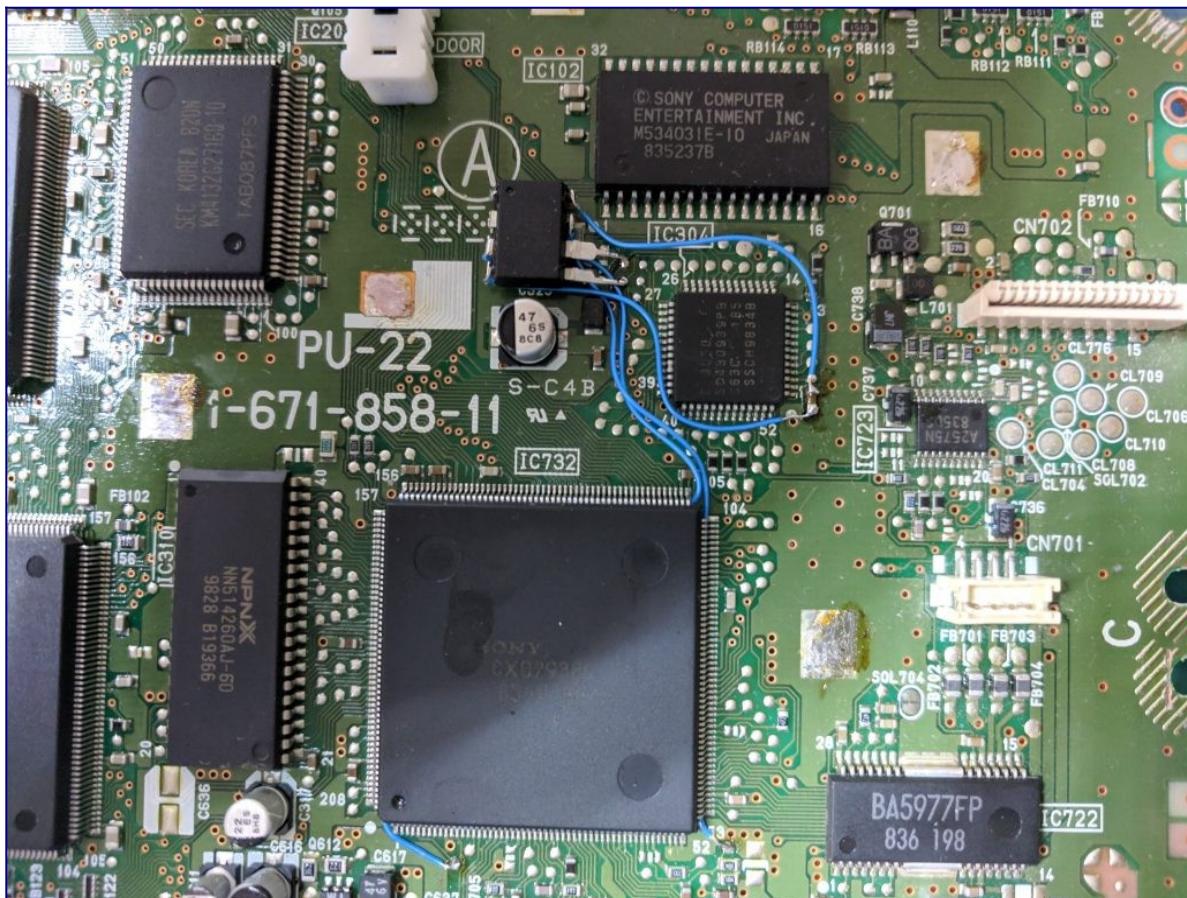
Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. William Quade (me)

## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

- Jon Co

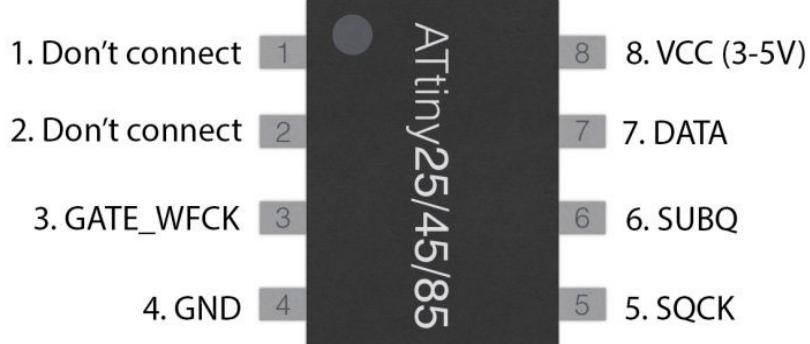


# PU-23 PsNee modchip installation diagram

The last version of the original PlayStation 1 design was the SCPH-900x. Most of these consoles have PU-23 boards inside. The most notable difference between the 9000 series and earlier consoles is that Sony removed the parallel IO port on the back. They also shrunk down the size of the board significantly.

## PU-23 PsNee modchip installation diagram

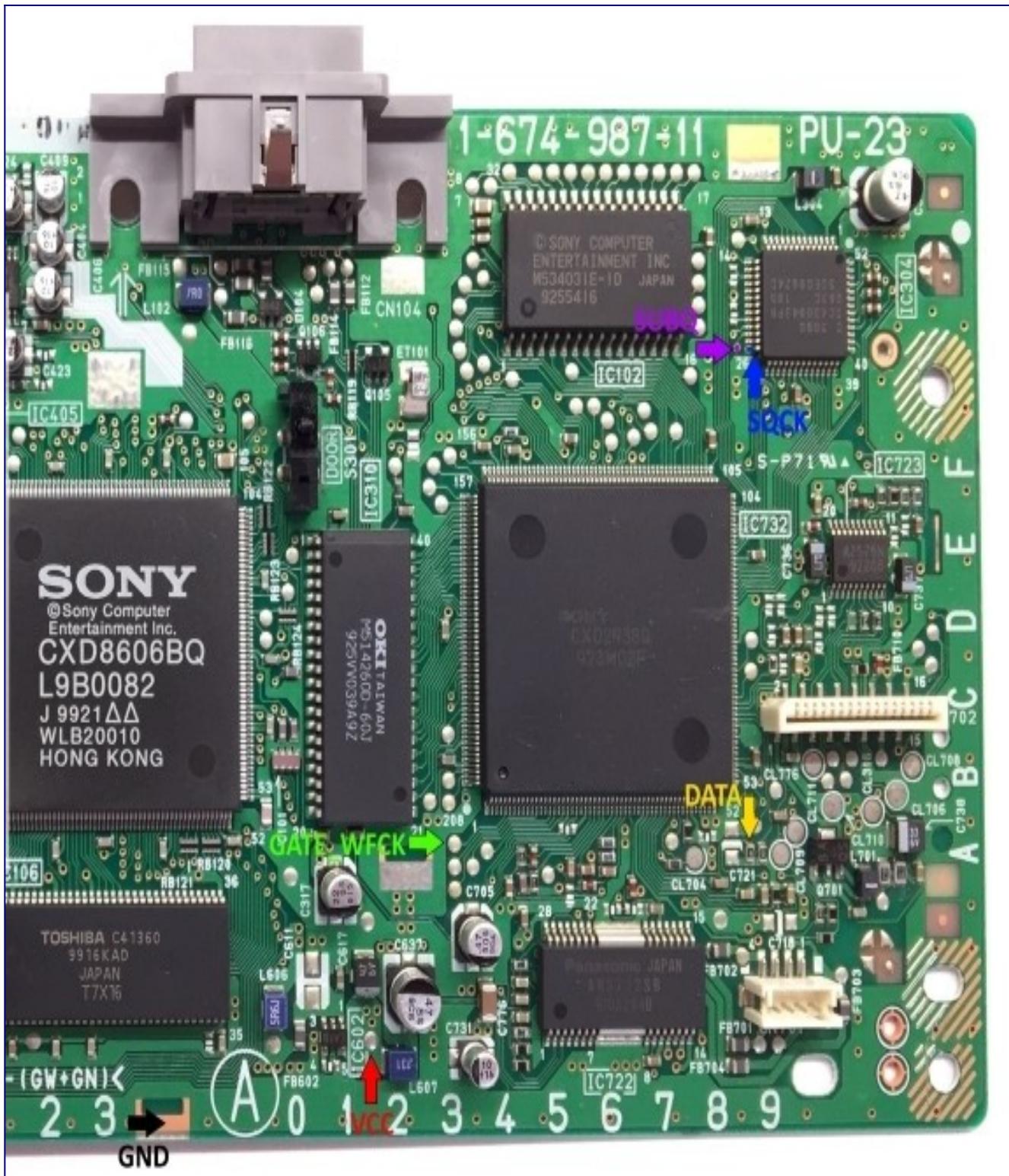
### ATtiny25/45/85 pinout

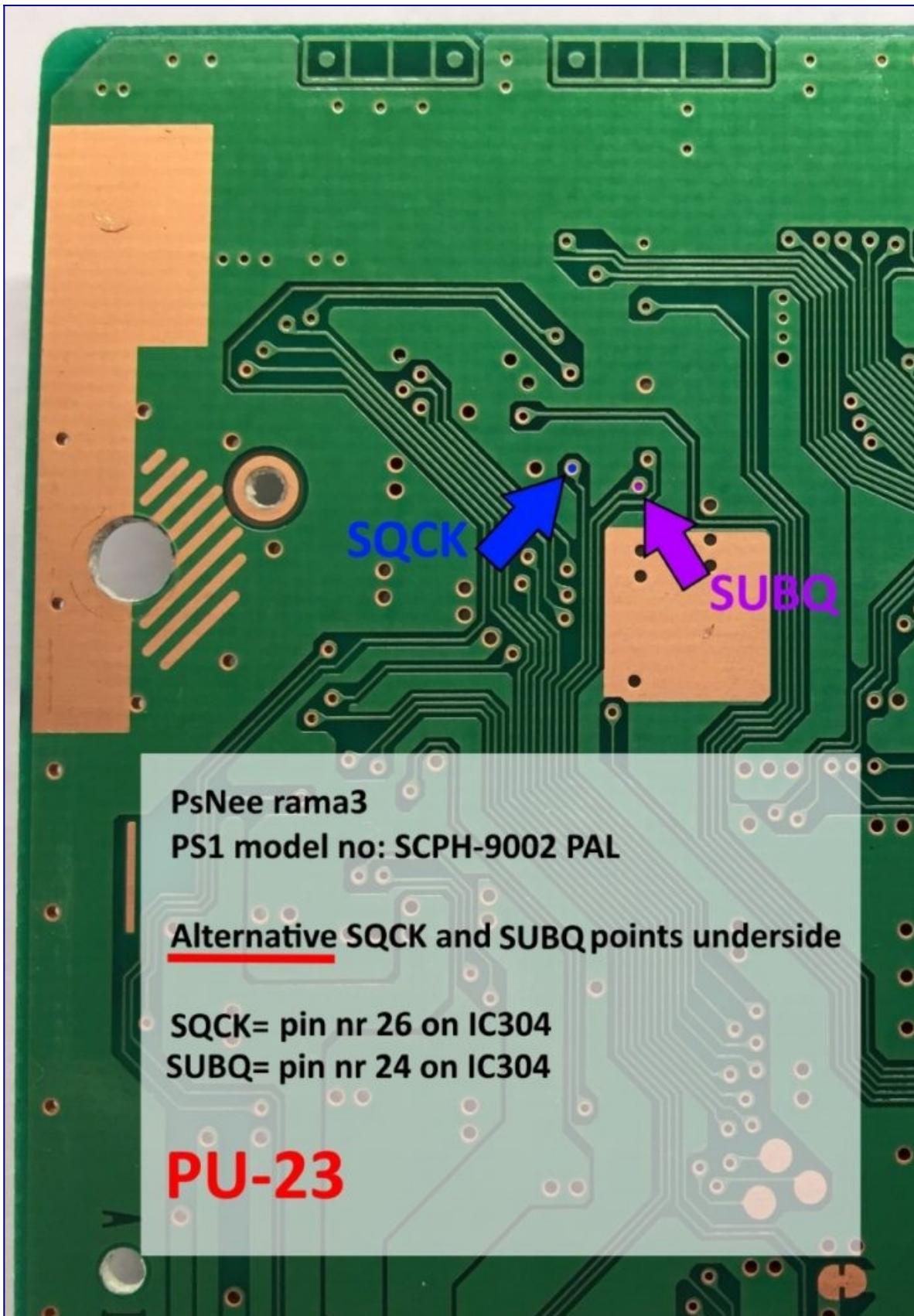


PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK





Above are some diagrams for installing the PsNee into a PU-23. The first diagram shows the pinout of the ATtiny45, the second shows the installation diagram (including pin numbers for Arduino

users), and the third shows some alternative points. Click on an image to open it in a new tab where you can zoom in to get more detail.

Let me know if any of the points aren't clear, I can clarify them for you.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PU-23.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.
- For DATA, be careful not to bridge the connection to the other resistor next to the one you are soldering to.
- Don't apply too much solder to SQCK, or you'll bridge the pins on the chip.
- For the wires that go into the holes, or vias, of the board: it's easiest to stick a small 30 AWG wire through the hole, then heat the wire and hole while adding solder.
- Placing the chip on top of the chip above SUBQ and SQCK is a good spot.

## Diagram success rate

Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. William Quade (me)

## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

1. William Quade (me)

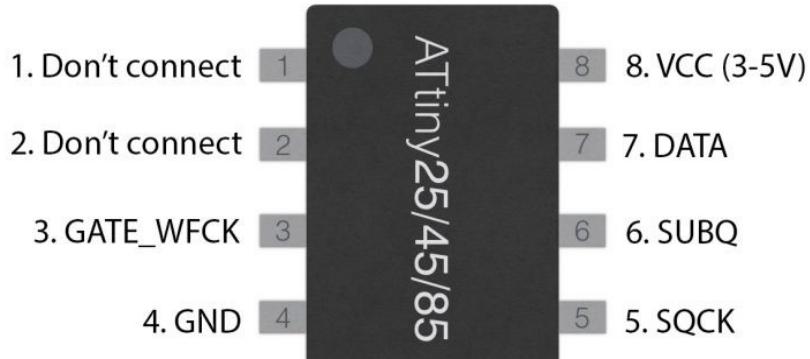


# PM-41 PsNee modchip installation diagram

Towards the end of the PlayStation 1's life Sony released a slim model called the PSone, or SCPH-10x. Many of these systems have PM-41 boards inside, which is what this page covers. Some of the boards are [PM-41 \(2\)](#) and require you to use a different diagram. The model is printed on your board if you take apart your console.

## PM-41 PsNee modchip installation diagram

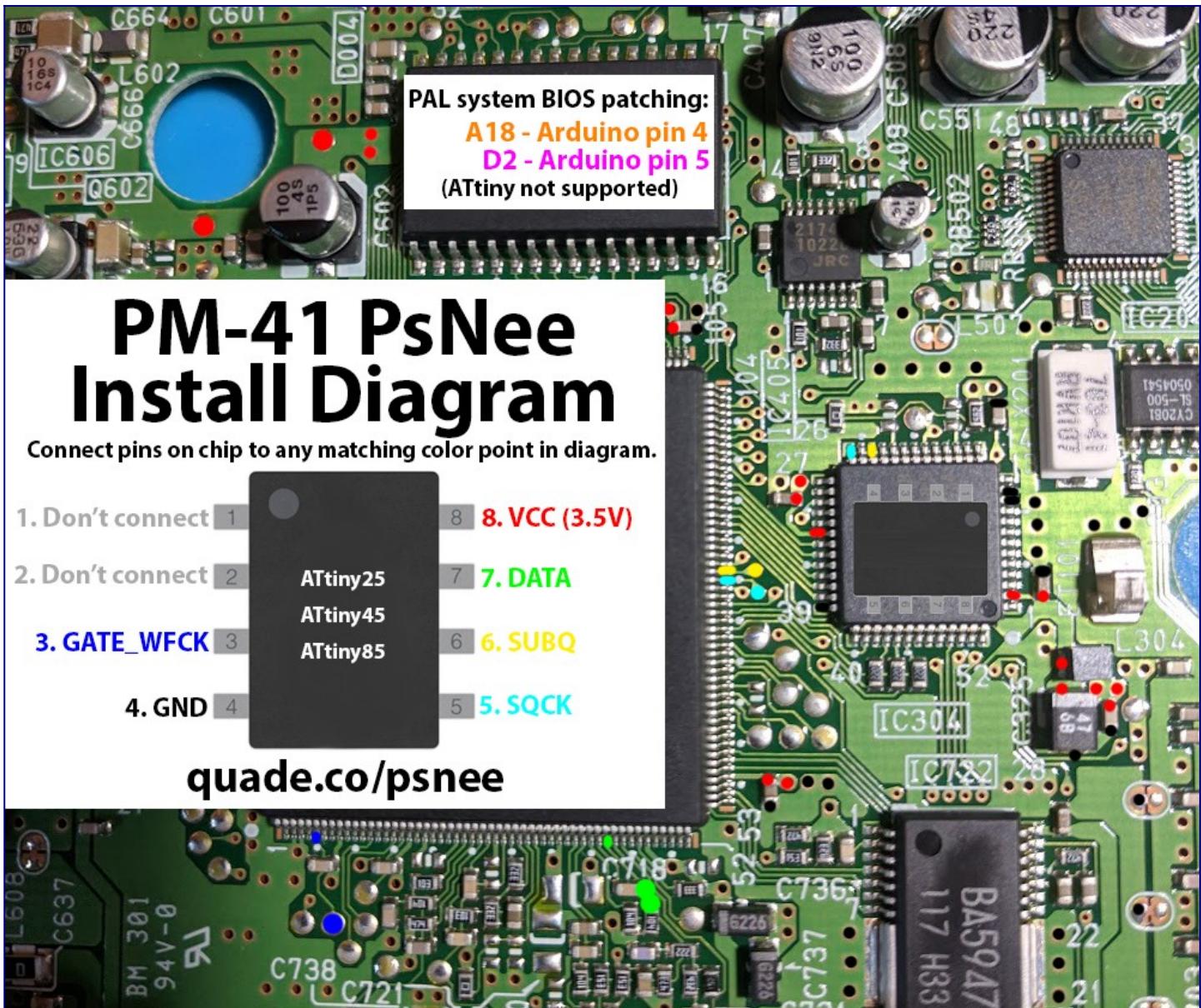
### ATtiny25/45/85 pinout

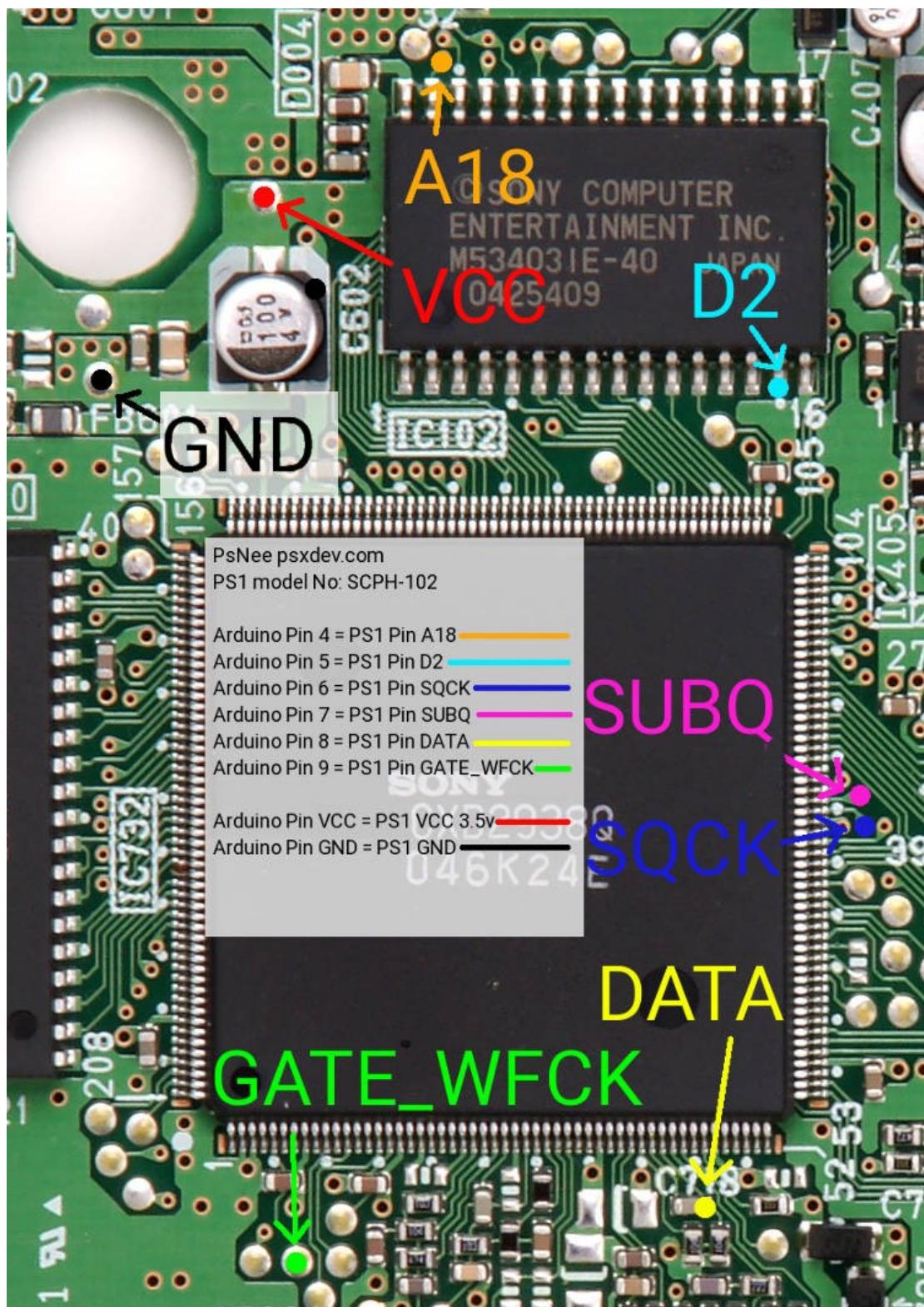


PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK





Above is the installation diagram. Just match each colored pin label in the diagram with any matching colored point on the board. There is a second image of the chip in the diagram showing a good place to position the chip.

If you have a PAL PSone system then you'll need to use an Arduino board and connect A18 and D2. Otherwise just ignore A18 and D2.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PM-41.

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- If you are using an ATtinyX5, or have a SCPH-101, then ignore pins A18 and D2.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.
- For the wires that go into the holes, or vias, of the board: it's easiest to stick a small 30 AWG wire through the hole, then heat the wire and hole while adding solder.
- For DATA, be careful not to bridge solder to a nearby component, or knock the capacitor off the board.

## Diagram success rate

Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. Nobody yet.

## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

- Nobody yet.

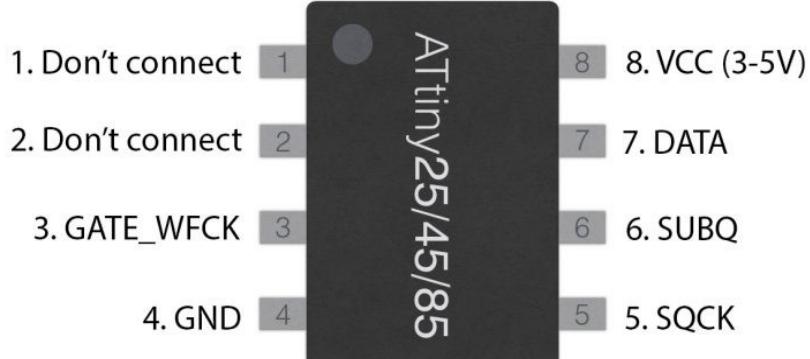
# PM-41 (2) PsNee modchip installation diagram

Towards the end of the SCPH-10x PSone production Sony revised the board slightly and released the PM-41 (2) board. This page covers the installation diagram for installing a PsNee chip into the PM-41 (2).

If your board has PM-41 printed on it then you should follow the [PM-41 guide](#) instead.

## PM-41 (2) PsNee modchip installation diagram

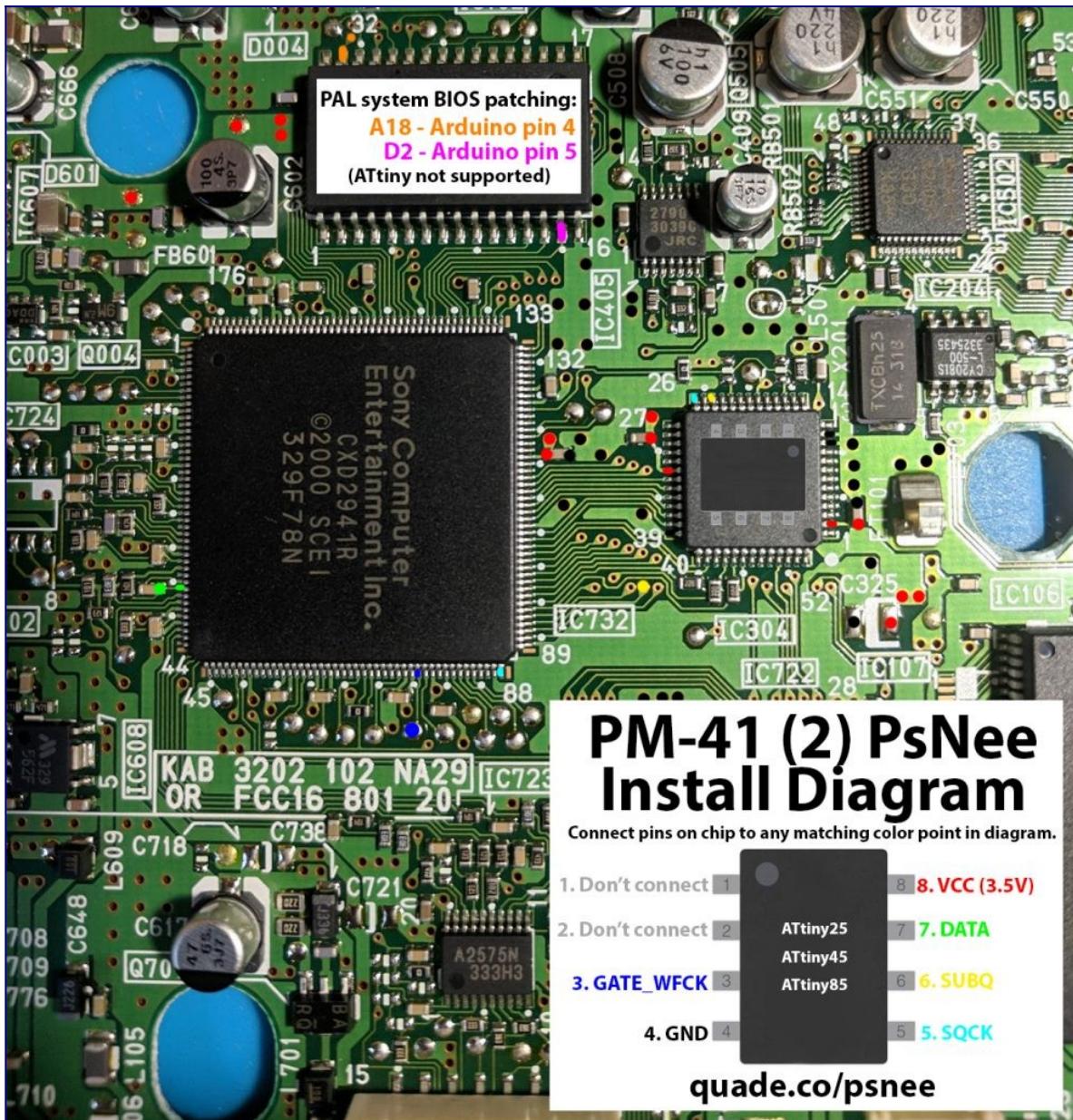
### ATtiny25/45/85 pinout



PsNee

Arduino pinout:

- Pin VCC – VCC
- Pin GND – GND
- Pin 3 – Debug TX
- Pin 4 – BIOS A18
- Pin 5 – BIOS D2
- Pin 6 – SQCK
- Pin 7 – SUBQ
- Pin 8 – DATA
- Pin 9 – GATE\_WFCK



Above is the installation diagram. Just match each colored pin label in the diagram with any matching colored point on the board. There is a second image of the chip in the diagram showing a good place to position the chip.

If you have a PAL PSone system then you'll need to use an Arduino board and connect A18 and D2. Otherwise just ignore A18 and D2.

## Installation tips

Here are some tips I have for you when you are soldering your chip into the PM-41 (2).

- **Cut your wires to be as short and direct as possible.**
- You don't need to connect pins one and two of the ATtinyX5 chip. Just desolder the wire.
- If you are using an ATtinyX5, or have a SCPH-101, then ignore pins A18 and D2.
- Use a multimeter to probe around for alternative VCC and GND points closer to where you position your chip for a cleaner installation.

- There may also be easier to solder to points for SQCT and SUBQ.
- Be super careful when soldering SQCT and SUBQ. It's easy to bridge pins if you aren't careful.
- Be careful with the DATA connection, it's easy to accidentally knock the capacitor off the board.

## Diagram success rate

Here's a list of people who have successfully modded their console with this diagram. Leave a comment and I'll add you to the list.

1. Nobody yet.

## Example installations

This section has photos of some successful installations which you can use to get a better understanding of how everything is wired and positioned.

Here are some sample installations I have done in SCPH-101's:

- Nobody yet.

# Ultimate PlayStation 1 modchip installation guide

I've been selling PS1 modchips on eBay for a while now. This post aims to be the ultimate source of information on PS1 modchips, including how to install them into every version of the PlayStation 1. I may have sent you here if you were looking for installation diagrams, or had questions about how to install the modchip. Continue reading and I'll answer those questions and more.

**As of right now this guide is still a work in progress.**

## **Guide structure**

This guide is structured into several pages. It's organized like this primarily so that comments can be more organized and useful for readers.

This page is the starting point of the guide. It includes some frequently asked questions about PlayStation 1 modchips in general, and then links off to more information about each type of chip, and those pages link off to installation guides for each console model.

## **Guide disclaimer**

I'm not a PlayStation 1 modchip expert. This guide has been put together from a combination of my experience, and what I have found online. Many of the sources for PlayStation 1 modding information have gone offline, or are spread out across many forum threads.

My goal is to have a single guide that covers as much information as possible, making it easier for everyone to get solid information about PlayStation 1 modchips. If you have any suggestions, corrections, or additional information, please leave a comment and I'll be happy to update this guide.

## **Modchip specific guides**

Here's a list of modchip specific guides, which include links to installation diagrams for each board, and source code to make your own chips.

- [MM3 modchip](#)
- [Mayumi v4 modchip](#)
- [ONEchip modchip](#)
- [PsNee modchip](#)

Modchips that aren't in the list above are covered in less detail in [a guide covering other older modchips](#).

## **What is a modchip, and what is it used for?**

The idea of a modchip is pretty straightforward. It is typically a small chip or board that is soldered to various parts of a video game console. The chip is programmed with special software that is used to circumvent DRM protection on the console.

In the case of the PlayStation 1 a modchip allows the console to do many new things. Here's a list of MM3 and Mayumi v4 modchip features.

- Play video game backups from any region.
- Play genuine games from any region.
- Works with all games, even the ones with anti modchip code.

A common use of a modchip is to play games from other regions, like Japan, that weren't released in the United States. With a modchip there is no need to import a console from each region.

## **PlayStation 1 modchip history**

The original PlayStation was one of the first consoles to have a modchip available for it. This was primarily due to the rise of cheap CD burners and blank discs that allowed game backups to be easily produced. That and the rising popularity of downloadable game backups online.

PlayStation 1 modchips were produced as early as 1996, not very long after the console was first released. These early mod chips were very expensive, but over time they were reverse engineered, cloned, and sold prices fell as a result. The original chips used the PIC16C54 microcontroller.

In 1997 a new modchip was released called "Old Crow". This new modchip was another clone of the original modchip, but the source code was released to the public, reducing the cost of the chip even further. "Old Crow" was originally made for the Zilog Z8 microcontroller, but versions were released for the PIC16C54, and PIC16C84 as well.

Later in 1997 the "Old Crow" code was ported to the then new PIC12C508 microcontroller, which is still being used for modchips today. The new chip dropped the modchip prices to the price range they are still sold at today.

Currently there are a few options for the PS1 modchip. The three most popular chips today are the MM3, the Mayumi v4, and the ONEchip.

All three are based off of the PIC12C508, but there are ports that allow the code to run on other chips like the PIC12F629. These chips are based on the "Old Crow" code with additional features like support for anti modchip games.

MM3 and Mayumi v4 chips are very similar, but operate slightly differently. Because of this some consoles may work better with one than the other.

ONEchip modchips are designed specifically for PAL region PSone's, and don't work with anything else. They were designed to bypass additional protection inside the SCPH-102.

There is also newer code designed to be used with modern Atmel processors like the ATmega and ATtiny series of chips.

## How does a PlayStation 1 modchip work?

PlayStation 1 modchips were designed specifically to bypass the copy protection and region coding that was built into game discs to prevent piracy.

In particular there is a stream of data sent from CD drive to the processor every time a game starts. The steam is sent in serial at 250bps and includes the text SCEI for Asian discs (NTSC/J), SCEE for European discs (PAL), or SCEA for North American (NTSC/US) discs.

The console's processor is programmed to look for this stream and only accept discs from its region. To bypass this copy protection mechanism the modchip removes the signal from the disc, and replaces it with the correct region coding. So for an American console the modchip will send the processor the SCEA string regardless of the disc that is inserted into the system is coded to send.

Anti modchip games have additional data in this stream to throw off some modchips. Newer chips like the MM3 check for this additional data and let it through to the processor if it exists, along with the spoofed region coding string.

PAL PSones have additional protection built into the BIOS. ONEchip and PsNee modchips connect to a couple BIOS pins and patch it to allow booting out of region games. Without the BIOS patch PAL PSones can only play PAL region games. Japanese systems starting with some SCPH-3000 models also have a similar region lock, but it hasn't been bypassed by any modchips yet.

If you want to learn more about how a PS1 modchip works, I recommend taking a look at the [PsNee project](#) on Github. The project is the modchip code for Atmel processors. The code is well documented and explains how everything works.

## What are the limitations of a modchip?

There are several problems you may run into with a PlayStation 1 with a modchip installed.

### 1. Japanese PS1 additional region lock

If you own a Japanese PS1, version SCPH-3000 or later (some SCPH-3000's are immune), and install a modchip into it, you may run into a problem when you try to load an American or PAL game. This is because starting with some SCPH-3000 PS1's Sony added an additional region lock to the BIOS of the system.

There are several ways to bypass this:

The first way is to replace the BIOS chip inside the Japanese PS1 with an American BIOS chip. Ideally you'd find one from the same model, for example if you have a SCPH-7500 you'd replace it with a SCPH-7501 BIOS chip. BIOS chips from other American models should work as well, but if you can get the same model there is less risk involved. This method requires a hot air rework station.

The second way is to replace the whole board inside your Japanese system with an American board. For example if you have a SCPH-7500 you'd put a motherboard out of a SCPH-7501 inside. This method would make your console appear to be Japanese from the outside, but it would really just be an American system.

The third way is to patch your game backups to appear like Japanese region games. This method is kind of a pain, especially if you already have a bunch of backups, but it doesn't require any additional hardware. You can use a [tool like this one](#).

At the moment there is no modchip that bypasses this region lock, but it is theoretically possible.

## 2. PAL and NTSC incompatibility with your TV

In North America we have NTSC-U/C consoles and TV's, and in Japan there are NTSC-J consoles and TV's. Meanwhile Europe and some other parts of the world there are PAL consoles and TV's. In the case of the PlayStation 1 the modchip can be used to unlock the console's ability to output video for any region based on the game being played.

The problem with this is that not all TV's accept all input signals. For example if you put a PAL game into your modded system and try to play it on an American TV you might not get any video on your TV.

To get around this problem I recommend buying a cheap [composite to HDMI converter](#). Typically these will take the weird analog signal that may be associated with a game from another region, and convert it into a nice and standard HDMI signal that your TV will display without any problems.

## 3. PAL PSone additional region lock

PAL PSones have additional region locking built into the BIOS. This prevents non-PAL games from loading, even if a modchip is installed. The good news is that there are several modchips that can patch the BIOS in order to bypass this protection. Both PsNee and ONEchip modchips have support for PAL PSone BIOS patching.

# What modchip should I choose for my PlayStation 1?

This is a question there is no perfect answer for. Each chip has its own pros and cons, and at this point in time there is no perfect chip that works 100% of the time with every PlayStation 1. I plan to do an experiment at some point in the future where I look more into the boot success rate of popular chips.

If you have a PAL SCPH-102 PSone then you need to either use a ONEchip, or an ATmega based PsNee (make sure whoever is selling it has enabled PAL PSone BIOS patching in the firmware as well). You really can't go wrong with either chip, just go with whatever is the easiest or cheapest to find.

If you have an older system, like a PU-7 or PU-8 SCPH-100x console, then you have a few options. One choice is the MM3 modchip, but from my experience they don't support full stealth mode with PU-7 boards, and aren't always the most reliable with every PU-8 board either. Stealth 2.8a chips are also available and work with PU-7 and PU-8 boards. What I would recommend is using a PsNee (either ATtiny or ATmega). I have confirmed that stealth mode works with the PU-7, and haven't ran into any reliability issues when booting games.

If you have any other version of the PS1 then you can use the MM3, Mayumi v4, or PsNee.

In general MM3 modchips are very popular and should at least partially function with every version of the PS1. Make sure you install at least seven wires if you want to be able to play anti modchip games.

Mayumi v4 modchips are also popular and should at least partially function with every version of the PS1 (except SCPH-100x). These chips require eight wires to be installed. They can be more reliable than MM3 chips in many cases because they are triggered using the PS1's internal clock and not the clock inside the PIC chip itself.

## **How can I make my own PlayStation 1 modchip?**

If you only need a few modchips you are probably better off buying your chips from me or another seller. However if you need a lot of modchips you might be able to save some money by buying the tools needed to make your own modchips.

There are three paths you can follow.

1. PIC12C508A based modchip (MM3, Mayumi v4, ONEchip)
2. Other PIC based modchip (MM3, Mayumi v4, or ONEchip code ported to work on another PIC model)
3. Atmel based PsNee modchip

### **PIC12C508A based modchip**

This is the traditional approach. The PIC12C508A is an older chip that's harder (and more expensive) to find a programmer for, and will reach end of production in the near future.

### **Other PIC based modchip**

This is an alternative approach. The PIC12F629 is a newer chip that's easier (and cheaper) to find a programmer for, and will continue to be produced for a longer than the PIC12C508A.

## **Atmel based PsNee modchip**

This is a more modern approach. I personally haven't tried doing this, but I can see it being the easiest option in the future. Arduinos are very popular, and they can be used as a modchip.

Standard Arduinos like the Arduino Pro Mini, and Arduino Uno can be used out of the box and programmed using any computer. Chips like the ATtiny45 are cheaper and can be used as well, but an external programmer is needed.

# Table des matières

Guide structure.....	1
PsNee modchip installation diagrams.....	2
PsNee modchip drawbacks.....	3
PsNee modchip advantages.....	3
PsNee modchip source code.....	3
PU-7 PsNee modchip installation diagram.....	4
PU-7 PsNee modchip installation diagram.....	5
Installation tips.....	6
Diagram success rate.....	7
Example installations.....	7
PU-8 PsNee modchip installation diagram.....	7
PU-8 PsNee modchip installation diagram.....	8
Installation tips.....	10
Diagram success rate.....	10
Example installations.....	11
PU-16 PsNee modchip installation diagram.....	13
PU-18 PsNee modchip installation diagram.....	14
PU-18 PsNee modchip installation diagram.....	14
Installation tips.....	16
Diagram success rate.....	16
Example installations.....	16
PU-20 PsNee modchip installation diagram.....	16
PU-20 PsNee modchip installation diagram.....	17
Installation tips.....	18
Diagram success rate.....	18
Example installations.....	19
PU-22 PsNee modchip installation diagram.....	20
PU-22 PsNee modchip installation diagram.....	21
Installation tips.....	22
Diagram success rate.....	22
Example installations.....	23
PU-23 PsNee modchip installation diagram.....	23
PU-23 PsNee modchip installation diagram.....	24
Installation tips.....	27
Diagram success rate.....	27
Example installations.....	27
PM-41 PsNee modchip installation diagram.....	29
PM-41 PsNee modchip installation diagram.....	29
Installation tips.....	30
Diagram success rate.....	31
Example installations.....	31
PM-41 (2) PsNee modchip installation diagram.....	31
PM-41 (2) PsNee modchip installation diagram.....	32
Installation tips.....	33
Diagram success rate.....	34
Example installations.....	34
Ultimate PlayStation 1 modchip installation guide.....	34
Guide structure.....	34

Guide disclaimer.....	34
Modchip specific guides.....	35
What is a modchip, and what is it used for?.....	35
PlayStation 1 modchip history.....	35
How does a PlayStation 1 modchip work?.....	36
What are the limitations of a modchip?.....	36
What modchip should I choose for my PlayStation 1?.....	37
How can I make my own PlayStation 1 modchip?.....	38
PIC12C508A based modchip.....	38
Other PIC based modchip.....	38
Atmel based PsNee modchip.....	38

With all our thanks to William Quade, for his permission to use his works of documentation <https://quade.co/>.