Quick Start

Getting Started

1. Plug in a Pico (you can solder it in, but more preferable to use SIL connectors, so that you can later unplug and swap with Pico Wireless for instance.
2. Plug in a USB cable to the Pico, and start coding :) No other connections or configuration needed. Some popular methods to work with the Pico are:

| C/C++ using Arduino IDE	MicroPython using Thonny	CircuitPython using Mu Editor	C/C++ SDK

3. It's not too easy to unplug the Pico! I use a blunt-edged plastic item (like a pen) and lever it out.

Using the Built-in Button and LED

Button B2 GPIO27	LED GPIO22

If the GPIO conflicts with any other circuit you are assembling, then you can unplug the shorting jumpers that are near the button and LED, to disconnect them.
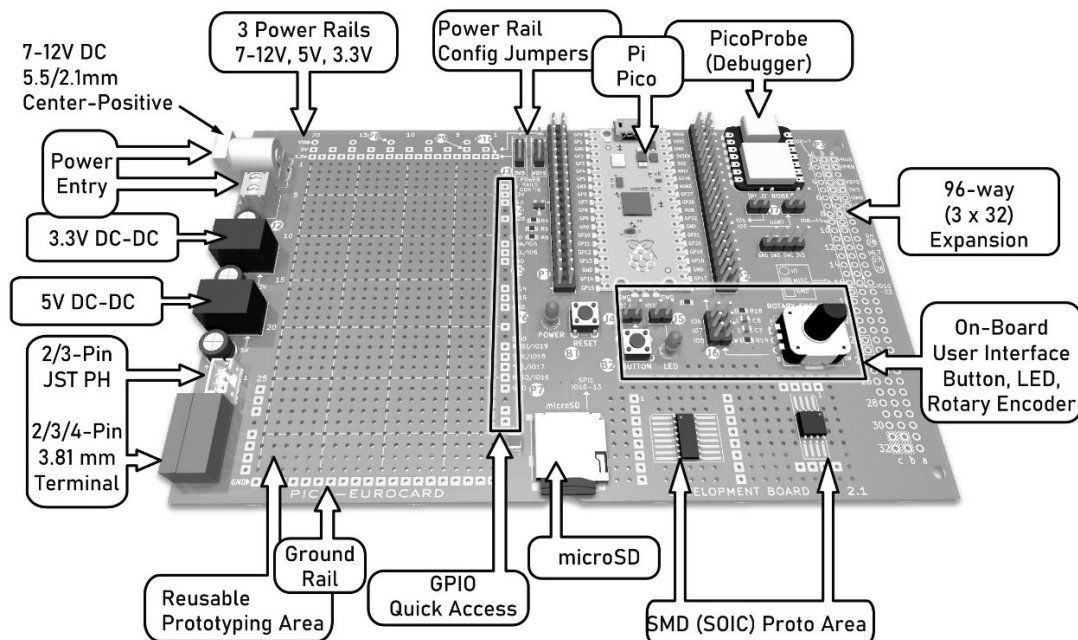
Using the Prototyping Area

1. The Power Rail Config Jumpers (labelled J3 on the board) are used to control the 5V and 3.3V power rails on the prototype area of the board. Here are the settings (there is a diagram in the For Reference section):
   a. When the jumpers are in the **lower** position, then the power rails are derived from the Pico USB connection. This is the normal setting.
   b. When the jumpers are in the **upper** position, then the rails are derived from an external power source (the Power Entry connectors on the left side of the board) via Voltage Regulators (DC-DC converters); this is described in step 2. You can continue to use the USB cable too; there is no harm in simultaneously connecting the supply and the USB cable.
2. If you want to power the board from a 7-12V source, then you can use the Power Entry connectors (J1 or J2), provided a 5V Voltage Regulator (DC-DC converter) is installed. The barrel connector is 5.5/2.1mm with **center-positive**. You can also optionally add a 3.3V regulator if you need more current than the Pico can provide. You can see the locations of the regulators on the left side of the board. As mentioned, set the Power Rail Config Jumpers to the upper position to set the prototyping area power rails to use the regulators.

3. Depending on preference, you could either solder parts to the prototyping area (they are easy to desolder and reuse the area), and just patch wires using either jumper cables or directly soldered wires, or you could connect a SIL connector to the GPIO Quick Access column (labelled P4,P5,P6,P7), and then plug on an external board, such as stripboard.
4. The GPIO Quick Access column provides access to some popular GPIO. The 5V and 3.3V supplies are also available on that column, to make it convenient to plug on a daughter board.

Using the Picoprobe Debugger

This section is only relevant for C/C++ SDK. You will need to install a 3-pin cable before you plug in the Pico, as described in the steps here:

1. The Picoprobe Debugger is very useful for working with the C/C++ SDK. It is not needed when working with Arduino IDE, or MicroPython or CircuitPython.
2. The Picoprobe Debugger is basically special firmware that runs on a separate RP2040 chip. For the Pico-Eurocard board, the XIAO RP2040 board is used. Solder the XIAO RP2040 onto the board, and then upload the Picoprobe firmware to it. It can be obtained here: https://github.com/shabaz123/rp2040tools
3. The Picoprobe connects to the target Pi Pico using a 3-wire connector. You can use the jumper cable supplied to do that. A jumper cable is used because different versions of the Pico (for instance the Pico W) have the connections in a different place on the Pico module.
4. Software development can be using the command line, or using an interactive development environment (IDE).
5. In terms of IDEs, you can use VS Code, or Jetbrains CLion to develop using the C/C++ SDK. Personally I find CLion easier, but others prefer VS Code. Both have their benefits/disadvantages.
6. To use VS Code, start with the blog here and follow the info/links: https://community.element14.com/products/raspberry-pi/f/forum/51735/first-experience-with-a-raspberry-pico---setup-toolchain-ide-debug-and-freertos
7. To use CLion the details are here: https://community.element14.com/products/raspberry-pi/b/blog/posts/using-clion-for-easier-coding-with-pi-pico-and-c-c
8. The Picoprobe has its own USB Type C connector. It can get messy having to connect both the target Pi Pico Micro USB cable, and the Picoprobe Type C USB cable both into the PC (unless you're actually developing a project that uses USB). You can actually power everything from the single Picoprobe Type C cable if you wish. To do that, connect a shorting jumper to the top-right of the board (labelled P3), to pins labelled VBUS and VPRB. **If you use the VBUS-to-VPRB shorting jumper, then you shouldn't connect up any other power supply**, and ideally you shouldn't connect both USB cables. It is best to disconnect the shorting jumper at all times unless you are powering everything from the Picoprobe Type C cable only. The VBUS/VPRB jumper is merely a convenience for the occasional times that you're only interested in coding, and are not powering up the board from other sources.

For Reference

Power Rail Config jumper diagram:
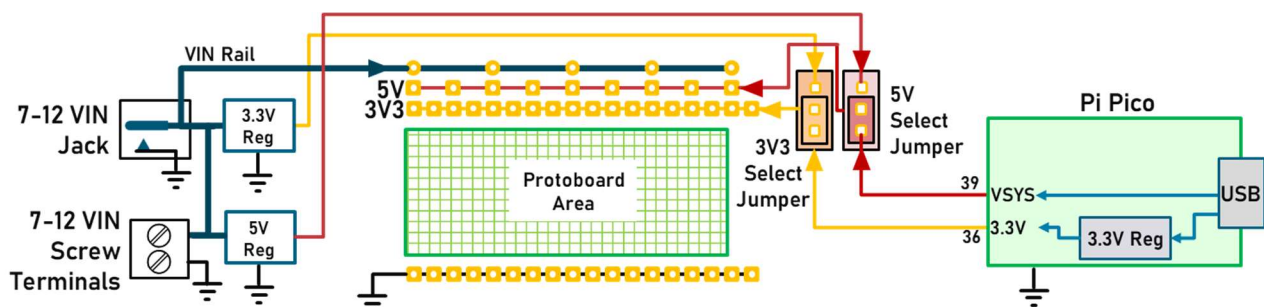
Further Information

Search for **pico_eurocard** on the element14 website, there are articles there covering the hardware as well as some projects.