**Pins**

Kicad files to make a custom Teensy 4.1, based on the official Teensy schematics, but exposing all pins for testing. All the 55 pins except D41 are brought out for testing

**Pins need for:**

* 1 SPI \*
* 4 PWM \*
* 3 CAN \*
* 10 GPIO \*
* 1 Power \*
* LED Pin \*
* Serial \*
* I2C \*
* Ethernet/ *Future implementation*
* Micro SD/ *Future implementation*
* LCD/ *Future implementation*

**Displays**

ILI9341 320x240 Color TFT

These displays are the best supported on Teensy 4.1, with multiple high-performance libraries for fast updates speed. ILI9341 is usually the best display to use, due to superior software support.

A diagram of a circuit board

Description automatically generated

A circuit board with wires connected to it

Description automatically generated

A table with text and symbols

Description automatically generated with medium confidence

PWM Timers

32 timers’ control PWM pins or may be used for other timing functions. Normally these timers are accessed with analog Write or libraries, but they have many very advanced features which may be accessed by direct hardware register use.

* **~~FlexPWM1 Module0~~**~~- Controls PWM pins 1, 44, 45.~~
* **~~FlexPWM1 Module1~~**~~- Controls PWM pins 0, 42, 43.~~ (Usados por CAN)
* **FlexPWM1 Module2** - Controls PWM pins 24, ~~46, 47~~.
* **FlexPWM1 Module3** - Controls PWM pins 7, 8, 25.
* **FlexPWM2 Module0** - Controls PWM pins 4, 33.
* **FlexPWM2 Module1** - Controls PWM pin 5.
* **FlexPWM2 Module2** - Controls PWM pins 6, 9.
* **FlexPWM2 Module3** - Controls PWM pins 36, 37.
* **FlexPWM3 Module0** - Controls PWM pin 53.
* **FlexPWM3 Module1** - Controls PWM pins 28, 29.
* **FlexPWM3 Module2** - No pins accessible.
* **FlexPWM3 Module3** - Controls PWM pin 41.
* **FlexPWM4 Module0** - Controls PWM pin 22.
* **FlexPWM4 Module1** - Controls PWM pin 23.
* **FlexPWM4 Module2** - Controls PWM pins 2, 3.

**Special Timers**

These extra timers allow delays, analog sample rate timing, carrier modulation, and other special timing tasks to be performed, without consuming any of the normal PWM-oriented timers.

* **GPT1** - Generic 32-bit timer

**CAN**

3 ports for CAN bus allow connecting to automotive & industrial control systems which use CAN communication. A CAN transceiver chip must be added to complete the electrical interface between Teensy 4.1 and the CAN bus. ([CAN1 pin 22 and 23], [CAN2 pin 0 and 1], [CAN FD pin 30 and 31])

**Digital Input Pins**

Digital pins may be used to receive signals. Teensy 4.1 pins default to INPUT most with a "keeper" resistor. Teensy 4.1 pins accept 0 to 3.3V signals. The pins are not 5V tolerant. Do not drive any digital pin higher than 3.3V.

**Power**

**USB Power**

Normally Teensy is powered by your PC or USB hub, through a USB cable. The USB power arrives at the VUSB pin, which is connected to VIN and powers the entire board.

**VIN Pin**

When USB power is not used, 5V power may be applied to the VIN pin. Because VIN & VUSB are connected, power should not be applied to VIN while a USB cable is used, to prevent the possibility of power flowing back into your computer. Alternately, a pair of pads on the bottom side may be cut apart, to separate VUSB from VIN, allowing power to be safely applied while USB is in use. (TODO: VUSB-VIN pads photo, right side)

**3.3V Power**

Teensy 4.1 has a voltage regulator which reduces the 5V VUSB / VIN power to 3.3V for use by the main processor and most other parts. Additional circuitry may be powered from the 3.3V pin. The recommended maximum for external 3.3V usage is 250mA. Teensy 4.1 is not meant to receive power on its 3.3V pin, but this can be done with special modification.

**LED Pin**

Pin 13 has an orange LED connected. The LED can be very convenient to show status info. When pin 13 is used as an input, the external signal must be able to drive the LED when logic HIGH. pin Mode INPUT\_PULLUP should not be used with pin 13.

**Serial**

8 serial ports allow you to connect serial devices, such as MIDI, GPS receivers, DMX lighting, ESP wireless modules, etc. All 8 serial ports are fully independent and can transfer data simultaneously. None are shared with USB (as is done on some Arduino boards). All 8 ports include FIFOs for better performance at high-speed baud rates. [SERIAL TX 20 and RX 21]

**I2C**

3 ports for I2C (signals SDA & SCL) allow connecting a wide variety of chips which use I2C communication. The Wire library is used for I2C. Each I2C chip connected to the same SDA/SCL wires needs a unique address. Multiple I2C ports allow you to easily use more than 1 chip with the same address. All I2C ports support 100, 400, and 1000 kbit/sec speeds.

**SPI**

3 ports for SPI (signals MOSI, MISO, SCK) allow connecting higher speed chips, SD cards, and displays which use SPI communication. The SPI library provides software support for SPI. The first SPI port features a FIFO for higher sustained speed transfers. Each SPI chip requires a chip select (CS) signal. Most libraries using SPI can use any digital pin. The SPI ports provide special hardware-controlled CS pins, which are used by specially optimized libraries for higher performance. ([SPI1 MOSI 26, SCK 27, CS 38, MISO 39],)