CH552 Multi-Protocol Programmer Module

DEVLAB FACTOR

Mfr. Part #: UE0090

Product Reference Manual (V1.0)

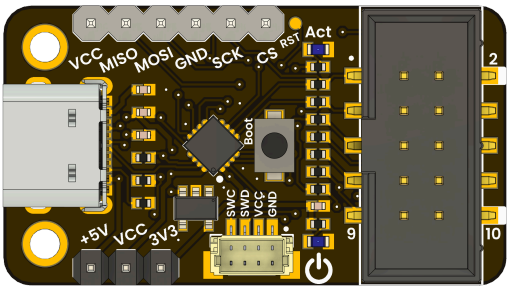
Description

The DevLab CH552 Multi-Protocol Programmer Module is a development platform that is both compact and versatile. It is specifically designed to facilitate the loading of firmware, the conversion of serial protocols, and the bridging of multi-architecture communication in embedded electronic systems.

The module is constructed around the WCH CH552 microcontroller, which is equipped with a high-performance 8051-compatible core that operates at a maximum speed of 24 MHz. The direct host communication is enabled by the integrated USB 2.0 full-speed interface, which supports native HID. This ensures a reliable and effective data exchange without the need for additional drivers.

This platform offers a unified solution for data-transfer applications, diagnostics, and programming, by providing comprehensive support for multiple communication standards, such as SPI, I²C, and UART. Hardware architecture facilitates the utilization of firmware implementations that are compatible with numerous embedded ecosystems, including:

- CMSIS-DAP protocol for ARM-based devices
- ISP/SPI programming for AVR microcontrollers
- General UART communication via USB-to-Serial conversion
- Interfaces that are interoperable with JTAG for the programming and debugging of CPLD/FPGA devices



The board operates from a single 5 V power supply provided through a USB Type-C connector. A selectable voltage output (3.3 V / 5 V) is available to ensure compatibility with a wide range of target microcontrollers and peripheral circuits. This configuration provides flexibility for both low-power and standard logic-level systems.

Prior to use, the module must be programmed with firmware supporting the desired communication protocol. Once configured, it functions as a reliable multi-protocol programmer or bridge interface, suitable for both standalone and integrated development environments.

The DevLab CH552 Programmer provides a cost-effective and efficient alternative to traditional programming tools due to its modular architecture and compact form factor. It is particularly well-suited for educational applications, rapid prototyping, and embedded system development, where flexibility and interoperability are essential.

Hardware Features

- **Microcontroller:** WCH **CH552G**, enhanced **E8051** core, up to **24 MHz**
- **Architecture:** MCS-51 compatible, single-cycle execution for most instructions
- **Memory:** 16 KB Flash (14 KB user + 2 KB BootLoader), 1 KB xRAM (DMA), 256 B iRAM, 128 B DataFlash
- **Clock Source:** Internal **24 MHz** oscillator with PLL
- **Power Supply:** Single **5 V** via **USB Type-C**, onboard **LDO (5 V → 3.3 V)**
- **Logic Voltage:** Selectable **3.3 V / 5 V** via jumper
- **USB Interface:** USB 2.0 Full-Speed with native **HID** support, Type-C host/device detection
- **Communication Interfaces:**
  – 2 × UART
  – 1 × SPI
  – Software-emulated I²C
- **Timers and PWM:** 3 × 16-bit timers with capture, compare, and PWM
- **Analog Interface:** Integrated **ADC**
- **Touch Interface:** Capacitive touch-key detection
- **Supported Protocols:**
  – **CMSIS-DAP** (ARM)
  – **ISP/SPI** (AVR)
  – **USB-to-UART bridge**
  – **JTAG-compatible** (CPLD/FPGA)
- **Connectors:** Protocol I²C port (Qwiic™/STEMMA™), standard expansion headers
- **Indicators:** Power and communication LEDs
- **Boot:** On-board boot-mode selection
- **Operating Voltage:** 3.3 V / 5 V (selectable)
- **Operating Frequency:** Up to 24 MHz
- **Form Factor: DevLab standard format**, compact modular PCB

Software Support:

- **Arduino IDE:**
  Fully compatible through the **UNIT Electronics CH552 Arduino Package**.
  Repository:
  https://github.com/UNIT-Electronics/Uelectronics-CH552-Arduino-Package
- **PlatformIO:**
  Supported for **advanced or experimental users**.
  Note: Not officially maintained by UNIT Electronics — recommended for experienced developers only.
- **CH55x SDK / Visual Studio Code:**
  Official SDK containerized with **Docker integration** for seamless cross-platform development (Windows, Linux, macOS).
  Repository:
  https://github.com/UNIT-Electronics-MX/unit_ch55x_docker_sdk
  Provides a **professional development environment** with **multi-board build** and **automated deployment** features.

DevLab Format Overview

The DevLab format is a standardized hardware layout that exposes the majority of the microcontroller or integrated circuit functional pins, including connections for external sensors, converters, and other peripheral devices. It provides direct access to I²C, SPI, UART, and additional communication interfaces for flexible hardware interaction.

Applications

- Embedded system programming and debugging
- USB-to-serial and protocol bridge development
- Multi-architecture firmware testing (ARM, AVR, CPLD/FPGA)
- Educational training and hardware prototyping
- Development of USB-based tools and automation interfaces

CONTENTS

1 The Board

1.1 Accessories

- 1×6-pin 2.54 mm Female Header – Primary interface header for serial, SPI, and I²C connections.
- 1×3-pin 2.54 mm Male Header – Configuration header used for voltage or mode selection.
- 1×1 2.54 mm (0.1 inch) Jumper Shunt – Removable shorting cap for selecting logic level

2 Ratings

2.1 Recommended Operating Conditions

Electrical characteristics when the microcontroller is powered with 5V

Parameter	Description	Min	Typ	Max	Unit
Vin	Input voltage to power on the module	**3.7**	**-**	**5.5**	V
Vil	Input low level voltage	**-0.4**	**-**	**1.2**	V
Vih	Input high level voltage	**2.4**	**-**	**Vin+0.4**	V
Vol	Low level output voltage	**-**	**-**	**0.4**	V
Voh	High level output voltage	**VCC-0.4**	**-**	**-**	V
Icc24M*	Total supply current when Fsys=24MHz	**8**	**11**	**-**	mA
Icc6M	Total supply current when Fsys=6MHz	**4**	**6**	**-**	mA

Icc750K	Total supply current when Fsys=750KHz	**2**	**3**	**-**	mA
Iin	The input current without pull-down resistor	**-5**	**0**	**5**	uA
Idn5	The input current with pull-down resistor	**-35**	**-70**	**-140**	uA
Iup5	The input current with pull-up resistor	**35**	**70**	**140**	uA
IUP5X	The input current with pull-up resistor from low to high	**250**	**400**	**600**	uA
Vpot	Power on reset threshold	**2.1**	**2.3**	**2.5**	V

**24MHz Fsys only can be used when the microcontroller is working with 5V.*

Electrical characteristics when the microcontroller is powered with 3.3V

Parameter	Description	Min	Typ	Max	Unit
Vin	Input voltage to power on the module	**2.8**	**3.3**	**3.6**	V
Vil	Input low level voltage	**-0.4**	**-**	**0.8**	V
Vih	Input high level voltage	**1.9**	**-**	**Vin+0.4**	V
Vol	Low level output voltage	**-**	**-**	**0.4**	V
Voh	High level output voltage	**VCC-0.4**	**-**	**-**	V

Icc12M*	Total supply current when Fsys=12MHz	**3**	**5**	**-**	mA
Icc6M	Total supply current when Fsys=6MHz	**2**	**4**	**-**	mA
Icc750K	Total supply current when Fsys=750KHz	**1**	**2**	**-**	mA
Iin	The input current without pull-down resistor	**-5**	**0**	**5**	uA
Idn5	The input current with pull-down resistor	**-15**	**-30**	**-60**	uA
Iup5	The input current with pull-up resistor	**15**	**30**	**60**	uA
IUP5X	The input current with pull-up resistor from low to high	**100**	**170**	**250**	uA
Vpot	Power on reset threshold	**2.1**	**2.3**	**2.5**	V

**Use the jumper bridge to select the microcontroller's operating voltage.*

3 Functional Overview

The **DevLab CH552 Multi-Protocol Programmer Module** is a compact and adaptable hardware interface designed to support a wide range of **firmware flashing**, **protocol conversion**, and **serial communication** tasks across multiple embedded architectures.

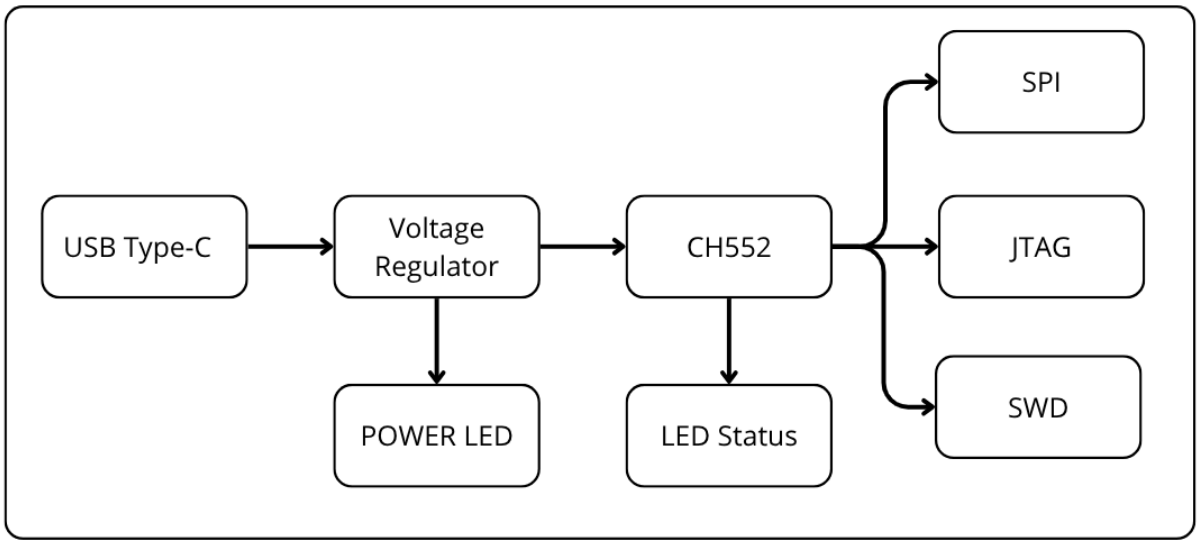
At its core, the module utilizes the **WCH CH552** microcontroller, which integrates a high-performance **E8051-compatible CPU**, **USB 2.0 Full-Speed** controller, and a flexible set of **SPI**, **I²C**, and **UART** interfaces.

This combination enables the DevLab CH552 to act as a **universal programmer and bridge device**, facilitating reliable data exchange and in-system programming for a variety of target platforms. The module supports several widely adopted programming and debugging standards, including **CMSIS-DAP** for ARM-based devices, **ISP/SPI** for AVR microcontrollers, and **JTAG-compatible** operations for CPLD and FPGA systems.

Designed with versatility in mind, the DevLab CH552 incorporates **on-board voltage regulation**, **USB Type-C connectivity**, and **selectable 3.3 V or 5 V logic levels**, allowing direct interfacing with both low- and standard-voltage systems. Its **standardized DevLab form factor** ensures ease of integration, rapid prototyping capability, and interoperability with other DevLab-compatible boards and sensor modules.

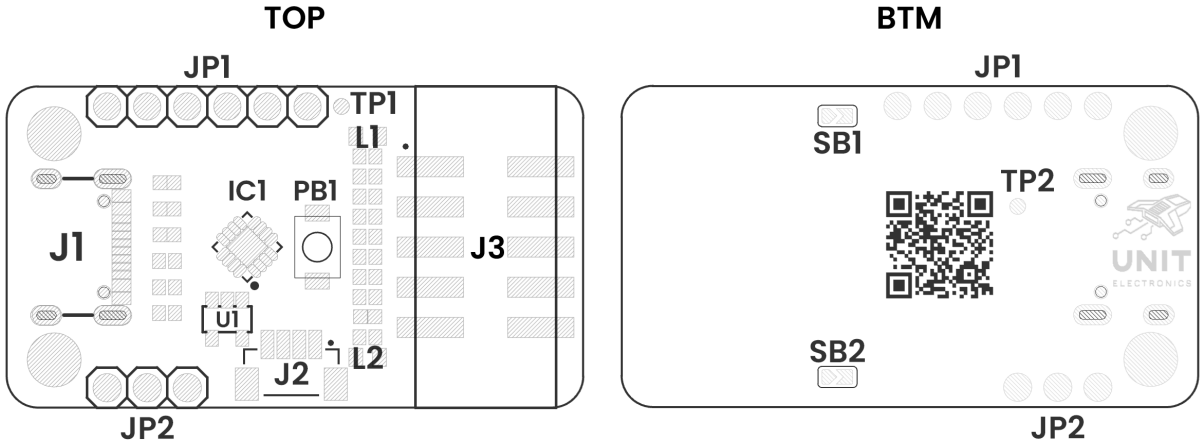
The DevLab CH552 Programmer Module is a cost-effective solution for educational laboratories, hardware testing environments, and embedded system development projects that necessitate cross-platform communication and programming flexibility. This is due to its compact footprint, multi-protocol support, and driverless USB HID operation.

Block Diagram



Block Diagram of DevLab CH552 Multi-Protocol Programmer

Board Topology



Views of Board Topology

Views of DevLab CH552 Multi-Protocol Programmer Topology

Table 3.2.1 - Components Overview

Ref.	Description
IC1	CH552 Microcontroller
U1	AP2112K 3.3V LDO Voltage Regulator
PB1	Boot Push Button
TP1	Reset Test Point
TP2	P3.1 Test Point
L1	Built-In LED
L2	Power On LED
SB1	Solder bridge to enable VCC at JTAG
SB2	Solder bridge to enable VCC at JST
J1	USB Type-C Connector
J2	Low-power I2C JST Connector
J3	JTAG Connector
JP1	Header for SWD or ICSP programming
JP2	Header to Select Operating Voltage Level

Microcontroller Overview — WCH CH552

The **CH552** is an **enhanced 8-bit microcontroller** based on the **E8051 architecture**, fully compatible with the **MCS-51 instruction set**. Approximately **79% of instructions** execute as single-byte, single-cycle operations, delivering up to **8~15 the performance** of a standard 8051. The core includes **dual data pointers (DPTR)** and optimized **xRAM data copy commands** for efficient data handling.

Designed for **low-cost and high-efficiency embedded applications**, the CH552 integrates a **USB 2.0 Full-Speed controller**, **SPI** and **dual UARTs**, enabling flexible communication options. The device features an **internal 24 MHz oscillator with PLL**, supporting stable and high-speed operation without the need for external components. It can operate from **2.8 V to 5.0 V**, making it suitable for both 3.3 V and 5 V systems.

With **14 KB of user Flash**, **1 KB xRAM**, and **256 B iRAM**, the CH552 provides sufficient resources for compact embedded projects. An additional **2 KB Bootloader area** allows in-system programming (ISP) via USB or UART. This microcontroller is widely used in applications such as **USB HID devices**, **serial converters**, **sensor modules**, and **rapid prototyping tools** due to its simplicity, performance, and cost efficiency.

CPU and Memory Features

- **Core:** Enhanced **E8051**, compatible with MCS-51 instruction set
- **Performance:** Up to 24 MHz system clock; single-cycle instruction execution
- **Program Memory:** 16 KB Flash (14 KB user + 2 KB Bootloader)
- **DataFlash:** 128 B non-volatile memory, byte-rewritable
- **RAM:** 256 B internal iRAM + 1 KB xRAM (DMA-capable)

Peripheral Features

- **USB:** Full-Speed USB 2.0 device controller with integrated transceiver and FIFO
- **UART:** 2 independent UARTs supporting high baud rates
- **SPI:** Master/Slave mode, built-in FIFO, clock up to ½ Fsys
- **I²C:** Software-implemented, compatible with standard-mode operation
- **ADC:** 4-channel, 8-bit analog-to-digital converter
- **Touch Interface:** 6-channel capacitive detection, up to 15 keys supported
- **Timers:** 3 × 16-bit timers (T0, T1, T2) with capture, compare, and PWM support
- **PWM:** 2 × 8-bit PWM channels
- **GPIO:** Up to 17 programmable I/O pins
- **Interrupts:** 14 interrupt sources (6 standard + 8 extended)
- **Watchdog:** 8-bit configurable watchdog timer

Clock and Power

- **Clock Source:** Internal 24 MHz oscillator with PLL; optional external crystal
- **Operating Voltage:** 2.8 V – 5.0 V
- **Low-Power Mode:** Sleep and wake-up from USB, UART, SPI, or GPIO

Additional Features

- Built-in **unique ID** for device identification
- **USB Type-C host/device detection** support
- **Multiple reset sources:** Power-on, software, watchdog, external reset pin

AP2112K and MCP73831 Power Management System

The **AP2112K** low-dropout (LDO) regulator IC provides a **stable and efficient power delivery system** for the DevLab BME688 Environmental Sensor Module. The board accepts input voltages of **up to 6V via Vcc**, and delivers a **regulated 3.3V output** with a nominal current of **350 mA** and peak support up to **600 mA**, protected by internal thermal shutdown.

Resistors for Signal Line Protection

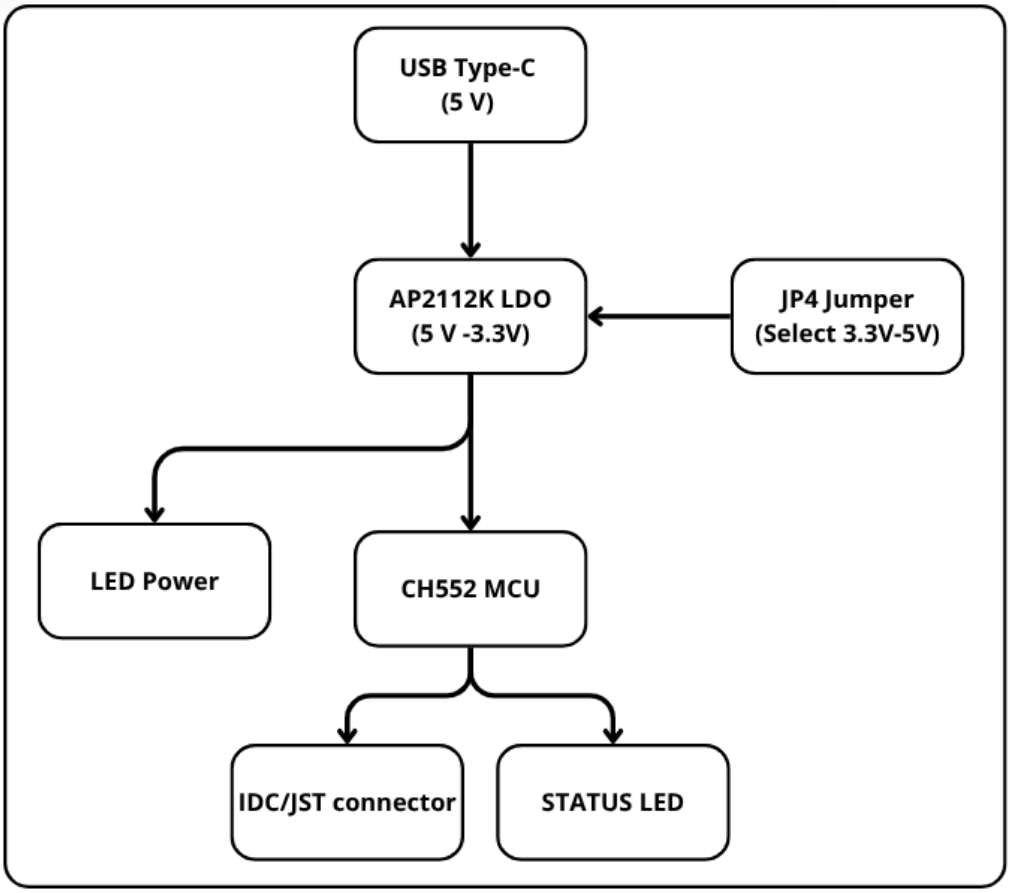
Function	Description
Purpose	Each communication line between the CH552G microcontroller and the IDC programming connector includes a 22 Ω series resistor.
Signal Conditioning	Acts as a damping element to reduce ringing, reflection, and EMI on high-speed lines such as SWD, SPI and UART.
Current Limiting	Restricts transient current when both the programmer and target device drive the same line, protecting I/O pins from potential damage.
Voltage Isolation	Improves tolerance between systems operating at 3.3 V and 5 V, ensuring stable and safe signal levels.
Result	Enhances signal integrity, system reliability, and cross-platform compatibility without affecting communication speed.

Power Tree

The DevLab CH552 Multi-Protocol Programmer Module receives power through a USB Type-C connector, which supplies a regulated 5 V input to the system. An integrated AP2112K low-dropout (LDO) voltage regulator steps down this input to 3.3 V, providing two stable power rails—3.3 V and 5 V—for internal and external use.

A **voltage-selection jumper (JP4)** allows configuration of the **CH552G microcontroller** and the external communication interfaces (**SPI** and **UART**) to operate at either **3.3 V** or **5 V** logic levels. The **IDC connector** includes a **resistive divider network** to maintain signal integrity and voltage reference accuracy, ensuring reliable interoperability with target devices operating at various logic standards.

This **power architecture** ensures complete electrical compatibility with both **3.3 V** and **5 V** systems, enabling **flexible integration**, **system evaluation**, and **rapid prototyping** across a wide range of embedded applications.



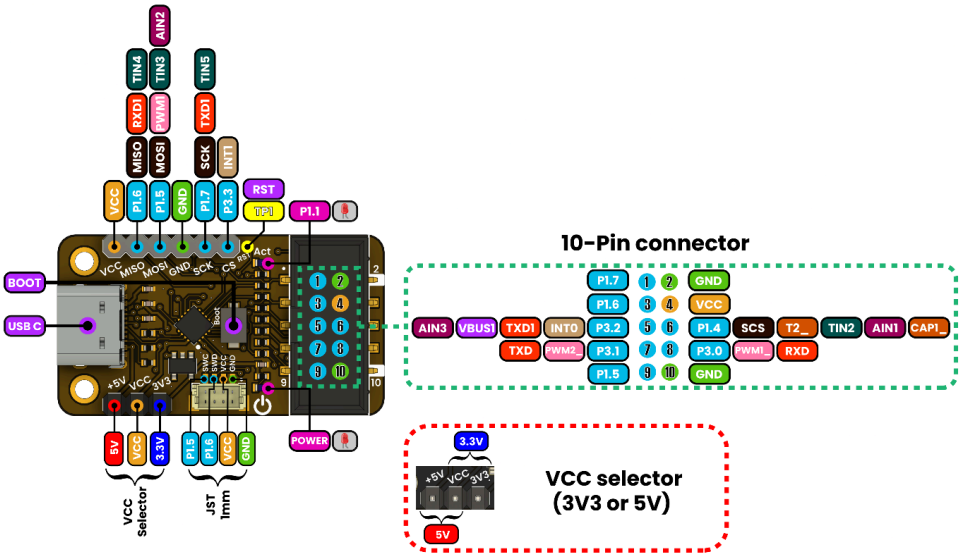
Power Tree

4 Connectors & Pinouts

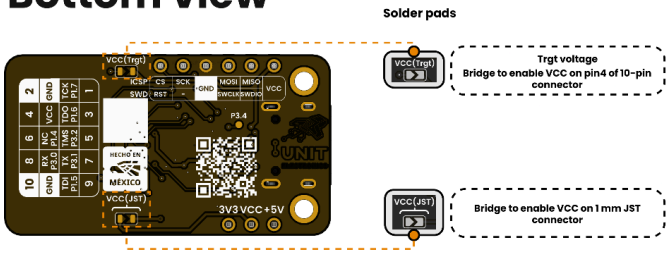
PINOUT

DevLab:CH552 Multi-Protocol Programmer Module

Top View



Bottom view

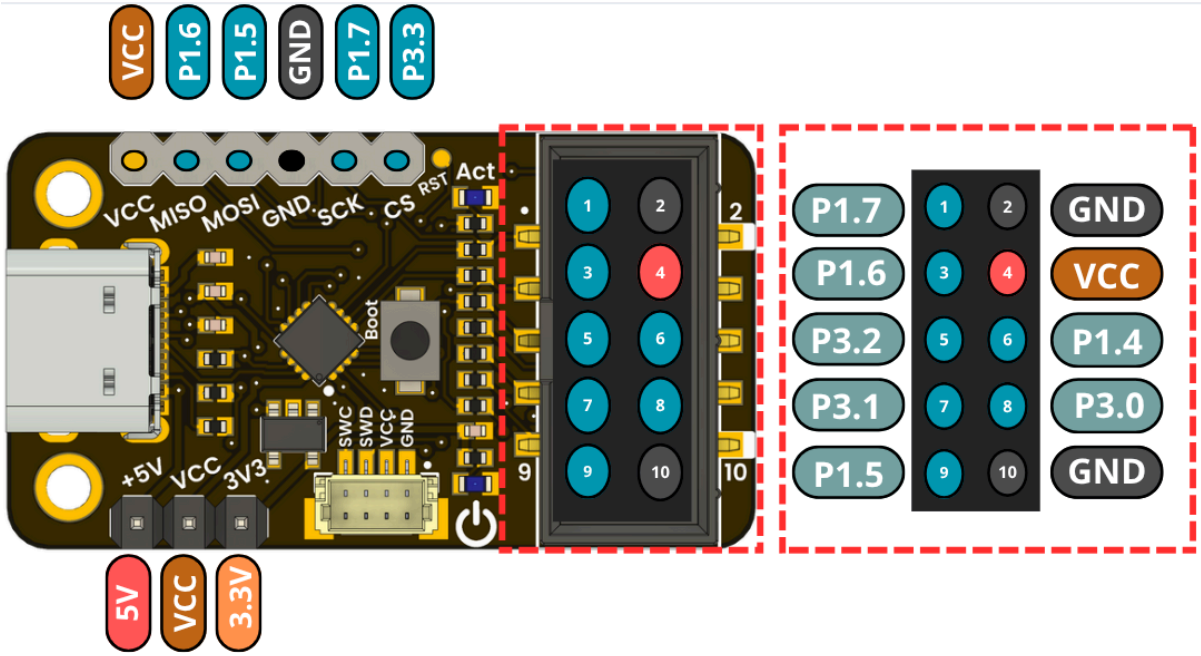


Leyenda

Module General Pinout

Pinout General Description

The DevLab CH552 Multi-Protocol Programmer integrates multiple communication interfaces within a compact form factor. Each interface is assigned to specific GPIO pins of the CH552 microcontroller, ensuring optimal functionality and minimal signal interference. The recommended distribution is summarized below.



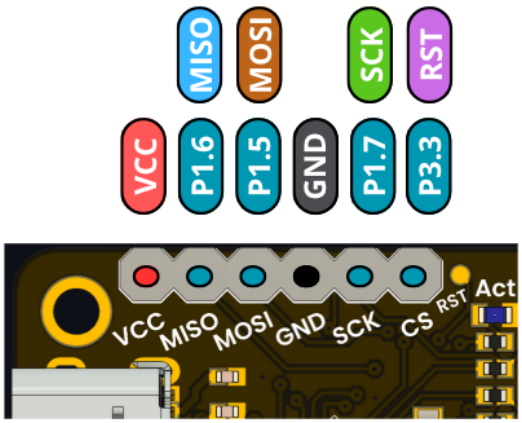
+

Interface	Description	Signals / Pins	Typical Use
JTAG	Standard boundary-scan and debug interface	TCK, TMS, TDI, TDO	Full chip programming, in-circuit test, debug
SPI	High-speed serial peripheral interface	MOSI, MISO, SCK, CS	Flash AVR Core programming, peripheral data exchange
SWD	ARM's two-wire serial debug and programming interface	SWCLK, SWDIO	Cortex-M programming & step-through debugging
JST Header	Compact connector for power + single-wire debug signals	SWC (SWCLK), SWD (SWDIO), VCC, GND	Quick-connect to target board for SWD and power

Configuration SPI

The **SPI controller** in the **DevLab CH552** includes an integrated **FIFO buffer** that enhances data throughput and reduces CPU load during high-speed operations. It supports clock frequencies up to **½ of Fsys (12 MHz at 24 MHz system clock)**, providing stable and efficient communication for in-system programming and peripheral data transfer.

Designed for flexibility, the controller supports **full-duplex and simplex modes** and can operate as either **Master** or **Slave**. This makes it suitable for a wide range of applications, including **AVR ISP** and **CPLD JTAG emulation**, where precise timing and high-speed data exchange are essential for reliable firmware programming and boundary-scan operations.

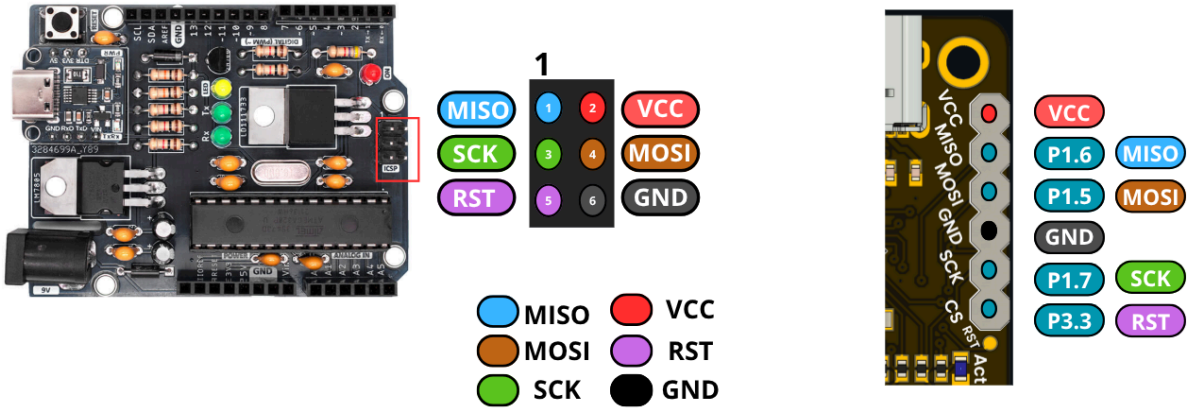


PIN	GPIO
MISO	1.6
MOSI	1.5
SCK	1.7
CS	3.3

Application: AVR ISP / SPI Programmer

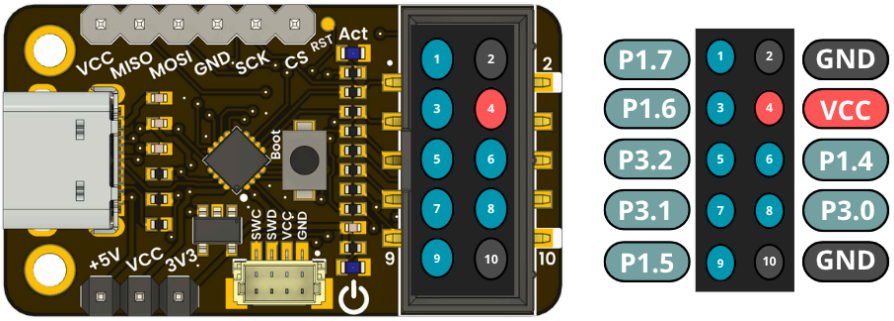
This configuration enables **in-system programming (ISP)** of **AVR microcontrollers** and **external SPI memory devices** through the CH552's hardware SPI interface operating in **master mode**. It supports reliable high-speed data transfer up to **12 MHz (Fsys / 2)**, allowing efficient flashing, EEPROM updates, and fuse configuration using standard AVR ISP signaling.

Only **AVR cores have been tested**, specifically **ATmega8**, **ATmega328**, and **ATtiny85**. The implementation is intended as a **test protocol** for validating programming and communication reliability rather than for production use. It serves as a functional demonstration of SPI-based in-system programming and timing compatibility between the DevLab CH552 and common AVR devices.



Distribution of connector IDC 10-Pin JST 2.4mm

The **IDC 10-Pin** and **JST 2.4 mm** connectors provide a unified hardware interface for multiple programming and debugging configurations. This distribution is designed to be **mechanically and electrically compatible** with standard **CPLD/FPGA JTAG programmers**, while also supporting **multi-protocol operations** such as **ARM CMSIS-DAP (SWD)**, **AVR ISP/SPI**, and **UART-based ESP8266 emulation**.



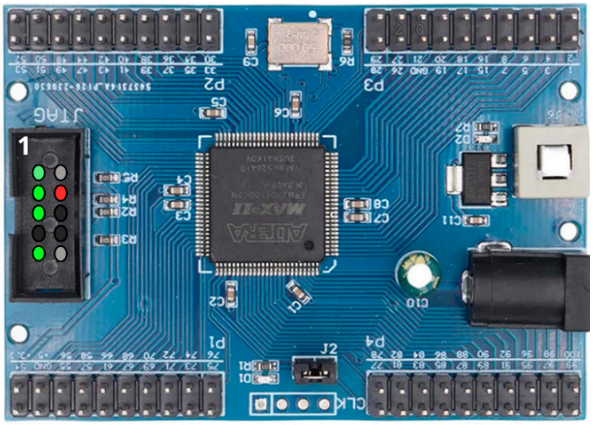
Through this connector, the DevLab CH552 exposes essential **serial, SPI, and general-purpose GPIO lines**, enabling flexible routing for different target architectures. By reconfiguring firmware or jumper selections, the same connector can be used to program microcontrollers, configure CPLDs, debug ARM devices, or communicate with peripherals — providing a compact and adaptable interface for development, testing, and production environments.

Application: CPLD / FPGA Programmer

This configuration enables **JTAG programming and functional testing** through a **standard 10-pin IDC connector**. It has been **validated with Altera/Intel MAX II (EPM240) CPLD devices**, employing a **direct one-to-one IDC pin mapping** to ensure clear, stable, and low-noise signal routing between the CH552 programmer and the target system.

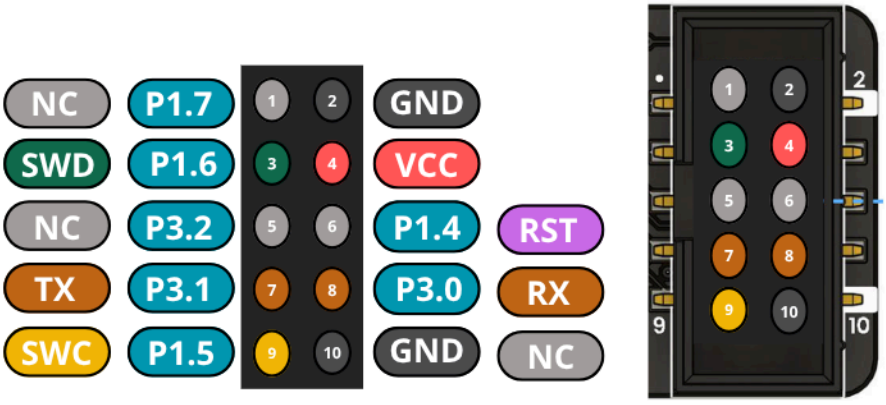
Firmware and Power Notes

The associated firmware is **open-source**, distributed under the **MIT License**, and intended for **educational, experimental, and development use**. For optimal operation, it is recommended to include a **VDD_TARGET solder pad** on the programmer board. Before connecting the IDC ribbon cable, always confirm **common ground reference** and **voltage compatibility** between the programmer and the target to prevent communication errors or potential hardware damage.



PIN	GPIO
TCK	1.7
TDO	1.6
TMS	3.2
TDI	1.5

Application: SWD Mode (CMSIS-DAP)



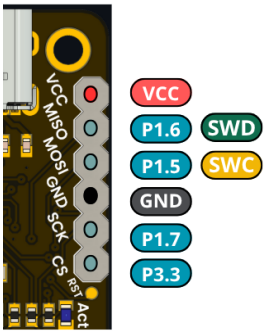
PIN	GPIO
SWDIO	1.6
SWCLK	1.5
RX	3.0
TX	3.1
RST	1.4

Application: ARM Cortex-M Programmer / Debugger

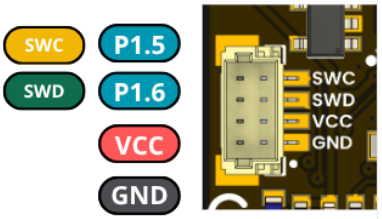
This configuration implements the **CMSIS-DAP firmware interface** to facilitate the **programming and debugging of ARM Cortex-M devices**. It also provides **UART pass-through functionality**, enabling serial communication for **real-time monitoring and target diagnostics**. The firmware has been thoroughly tested for **pyOCD compatibility** and validated across several ARM-based microcontrollers, including **PY32F003**, **RP2040**, **STM32F103**, **STM32F401**, and **nRF52840**. These tests confirm reliable operation in both **SWD (Serial Wire Debug)** and **UART bridge** modes. Proper **pin mapping** and **voltage configuration** for each target device are required to ensure stable communication. The programmer includes a **standard SWD header (SWC, SWD, VCC, GND)** for direct and secure connectivity with most Cortex-M development boards.

Notes and Recommendations

The CMSIS-DAP implementation is based on an **open firmware architecture**, which may require **specific firmware builds or configurations** for different target families. To maintain signal integrity, it is recommended to use **short cable lengths (<20 cm)** between the programmer and the target device. Always verify **voltage compatibility (3.3 V / 5 V)** and ensure a **common ground connection** before establishing communication to prevent data errors or hardware damage.



PIN	GPIO
SWDIO	1.6
SWCLK	1.5



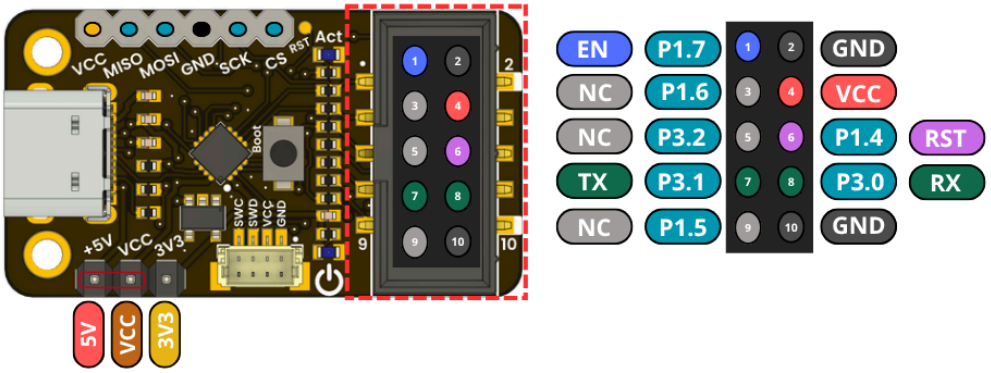
PIN	GPIO
SWDIO	1.6
SWCLK	1.5

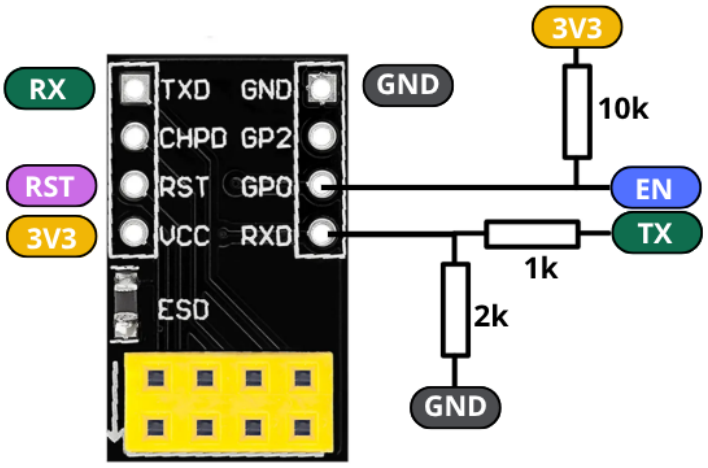
Application: Protocol UART

The **DevLab CH552 Multi-Protocol Programmer** can operate as a **USB-to-UART converter** for serial bootloading, debugging, or general device communication. In this configuration, the CH552's UART interface is fixed at a **non-configurable baud rate of 115 200 bps**, which ensures stable operation for most embedded devices and firmware flashing tools.

Although the CH552 runs at **5 V / 24 MHz**, target devices such as the **ESP8266** require **3.3 V logic levels**. To ensure compatibility, it is recommended to use **open-drain signaling** on the CH552 TX line together with an **external resistor divider** (**1 kΩ / 2 kΩ**) to reduce the output voltage to 3.3 V. A **10 kΩ pull-up** to 3.3 V must be connected to the **GPIO0 (EN)** line on the ESP8266 to maintain correct boot-mode behavior during reset.

This setup provides safe bidirectional UART communication when both devices share a common **3.3 V rail**, while the CH552 remains powered by **5 V from the USB Type-C port**.
It should be noted that this configuration offers **basic UART bridging only** and is **not recommended for new designs**, as it lacks isolation and advanced level-shifting features.





Signal	CH552 GPIO	ESP8266 Pin	Notes
TX	P3.1	RX	Use 1 kΩ / 2 kΩ resistor divider for level adaptation
RX	P3.0	TX	Direct connection (3.3 V tolerant)
RST	P1.4	RST	Open-drain output or via 10 kΩ pull-up
EN (GPIO0)	P1.7	EN	Pull-up 10 kΩ to 3.3 V
VCC	—	3.3 V	Shared supply for target device
GND	—	GND	Common ground reference

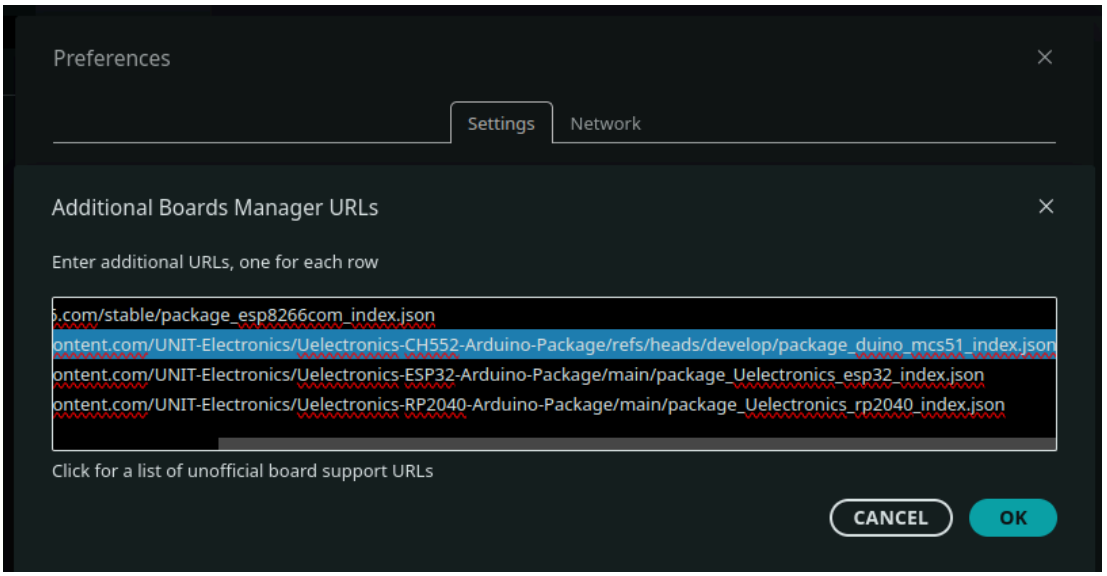
5 Board Operation

5.1 Getting Started with Arduino IDE

To configure the DevLab Multi-Protocol Programmer in the Arduino IDE (version 0.0.5 or higher is recommended), follow these steps:

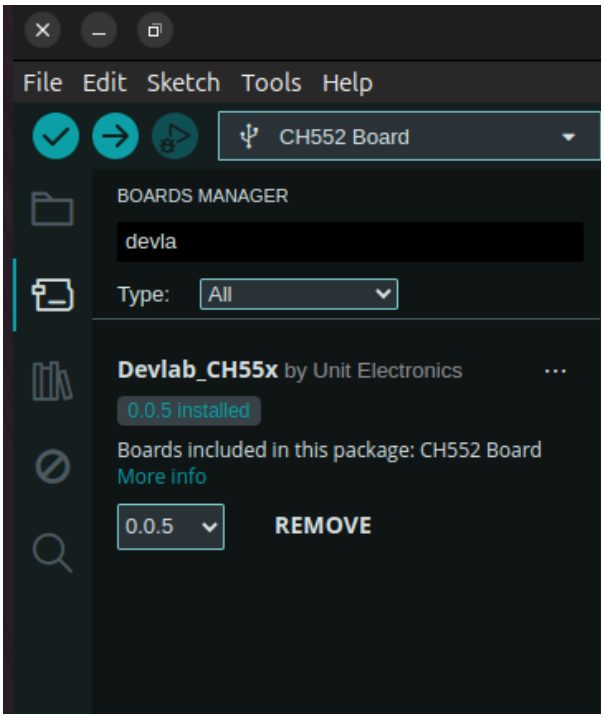
1. Adding the UNIT Electronics Boards Package:
    - **Open Preferences**, go to the **File** menu and select **Preferences** (or **Settings** on some systems).
2. **Add Board Manager URLs**:
    - In the **Additional Board Manager URLs** field, add the following links:
        - **CH552 JSON URL**:

          https://raw.githubusercontent.com/UNIT-Electronics/Uelectronics-CH552-Arduino-Package/refs/heads/main/package_duino_mcs51_index.json
    - Click **OK** to save.
3. **Tip**: If there are multiple URLs to add, separate them with commas.



4. Installing the Board Packages

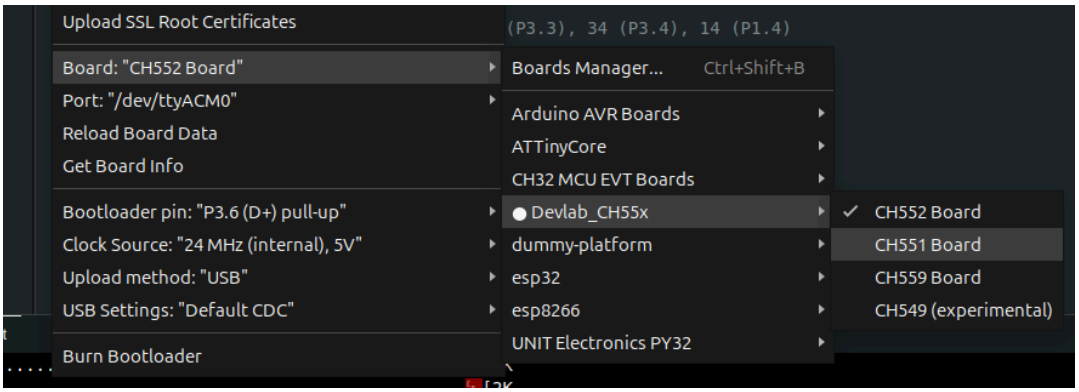
- **Open Board Manager:**
  Go to **Tools > Board > Board Manager**.
- **Search for UNIT Electronics Boards:**
  In the search bar, type **UNIT Electronics** .
- **Install the Latest Version:**
  Select the latest available version and click **Install**.



Note: Make sure you have an active internet connection during installation.

4. Selecting the Board

- **Set Your Board**: Go to **Tools > Board and** select the **Devlab** category. Once inside the selected category, choose the **CH552 board**.

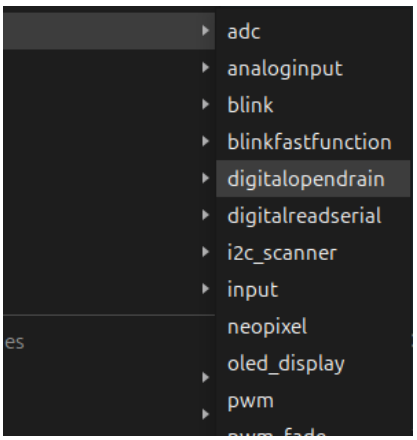


5. **Connect the Board**:

- Use a USB Type-C cable to connect the CH552 Programmer Module to your computer.
- Before connecting to the cable, press and hold the BOOT button, then plug in the USB connector while keeping the button pressed.
- The device will enter bootloader mode, allowing firmware to flash for approximately 5 to 10 seconds.
- If no programming operation occurs within this period, the CH552 automatically exits the bootloader and starts executing the main application code.

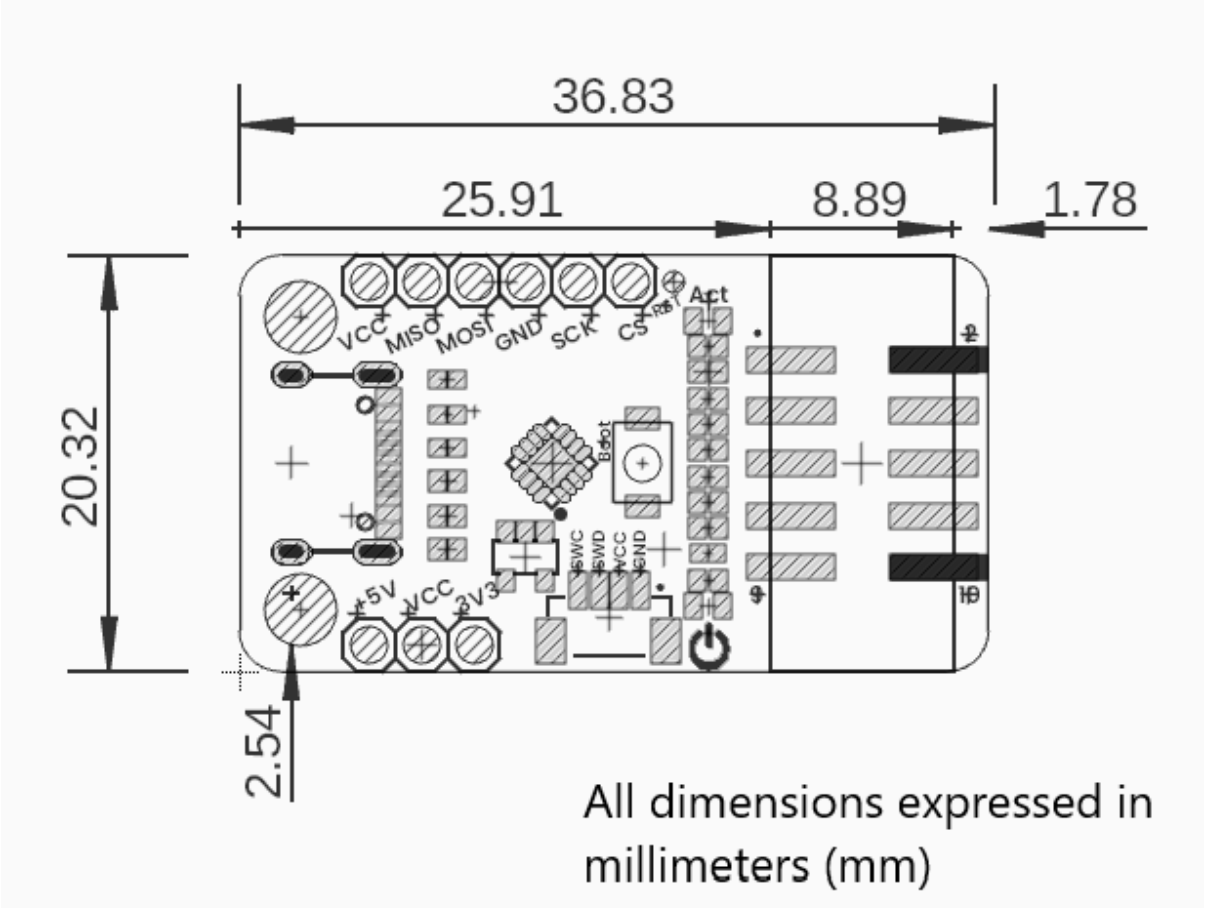
6. **Getting Started with Examples:**

- To start programming using the **Arduino IDE**, go to **File > Examples**.
  You'll find a wide variety of example sketches tailored for the board—perfect for exploring its capabilities and getting started with your first projects.



“By completing these steps, you’ll be ready to start programming your DevLab CH552 Multi-Protocol Programmer Module *using Arduino IDE”.*

6 Mechanical Information



Mechanical dimensions in millimeters

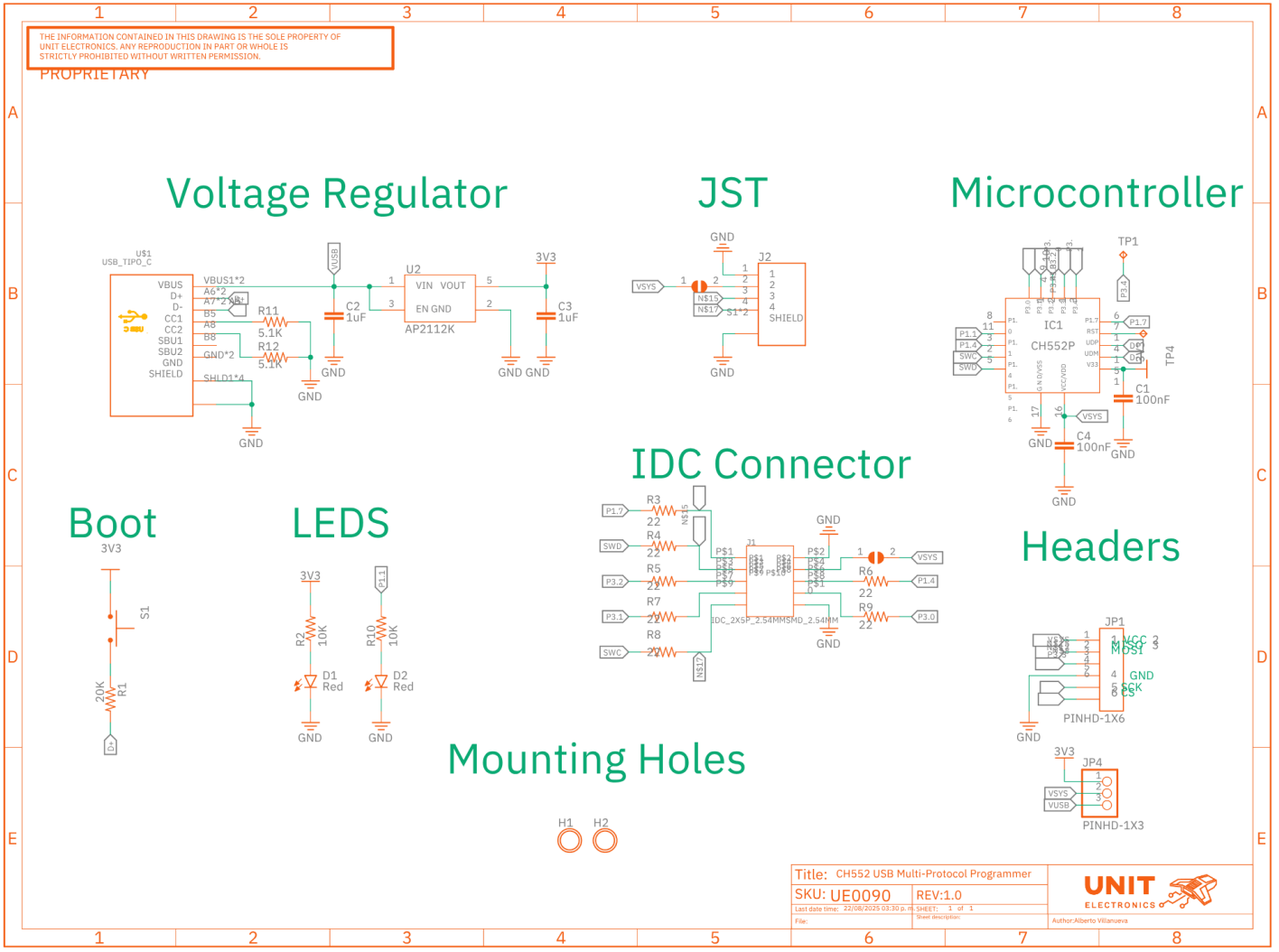
7 Company Information

Company name	UNIT Electronics
Company website	https://uelectronics.com/
Company Address	Salvador 19, Cuauhtémoc, 06000 Mexico City, CDMX

8 Reference Documentation

Ref	Link
Wiki	https://wiki.uelectronics.com/wiki/devlab-ch552-multi-protocol-programmer-module
Documentation (Getting Started)	https://unit-electronics-mx.github.io/unit_devlab_ch552_multiprotocol_programmer_module/
Github - Documentation	https://github.com/UNIT-Electronics-MX/unit_devlab_ch552_multiprotocol_programmer_module
CH552 Docker SDK	https://github.com/UNIT-Electronics-MX/unit_ch55x_docker_sdk
Arduino IDE	https://www.arduino.cc/en/software
Visual Studio Code	https://code.visualstudio.com/download

9 Appendix 9.1 Schematic (*https://github.com/UNIT-Electronics-MX/unit_devlab_ch552_multiprotocol_programmer_module/*)


THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF UNIT ELECTRONICS. ANY REPRODUCTION IN PART OR WHOLE IS STRICTLY PROHIBITED WITHOUT WRITTEN PERMISSION.
PROPRIETARY
Voltage Regulator
JST
Microcontroller
IDC Connector
Boot
LEDS
Headers
Mounting Holes
AP2112K
CH552P
IDC_2X5P_2.54MMSMD_2.54MM
PINHD-1X6
PINHD-1X3
Title: CH552 USB Multi-Protocol Programmer
SKU: UE0090
REV:1.0
Author:Alberto Villanueva