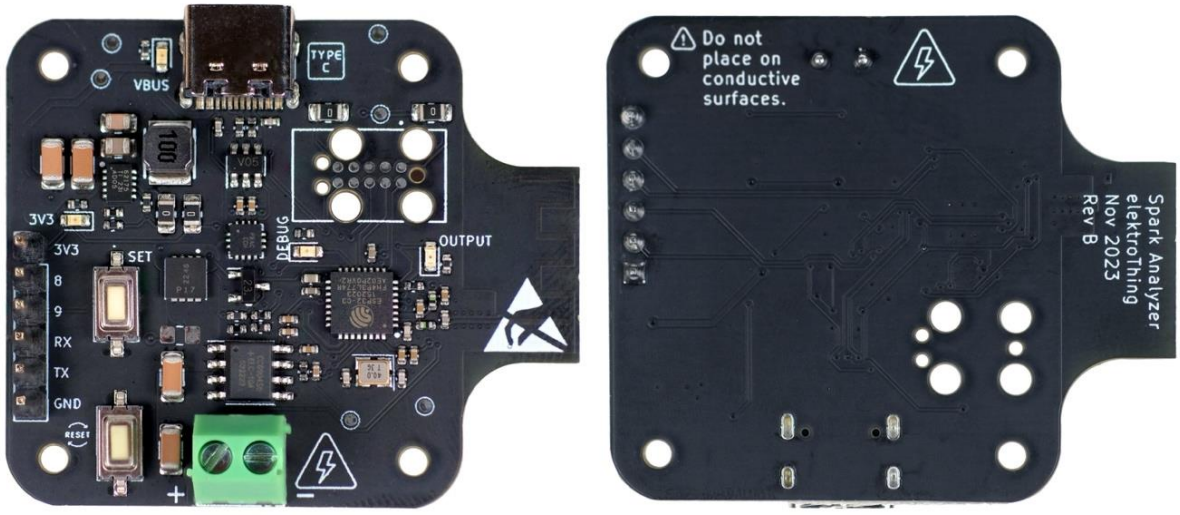
SPARK ANALYZER

Datasheet

Part No.: SPARKANALYZER-01



DESCRIPTION

The Spark Analyzer is a programmable USB-C Power Delivery Analyzer and Power Supply powered by an ESP32 microcontroller. It offers advanced features for power control and analysis, including Programmable Power Supply (PPS) and Power Delivery (PD), along with wireless control via Wi-Fi and BLE. Designed for flexibility and convenience, it is ideal for developers seeking precise power management and data logging capabilities in their projects.

SPECIFICATIONS

- Power (USB Type-C)
  - Programmable Power Supply (PPS): 3.3 to 21 VDC, up to 3.0A. Control at 20mV and 50mA resolutions.
  - Power Delivery (PD): Options of 5 VDC, 9 VDC, 12 VDC, 15 VDC, 20 VDC; max 5 A (100 W at 20 VDC).
  - ON Semiconductor FUSB302MPX: Programmable USB Type-C control and USB PD communication.
  - ESD Protection: On D+/D-/CC1/CC2 pins.
  - Texas Instruments TPS62175DQCT: 3.3 VDC 0.5 A max output DC-DC Step-Down Converter.
  - Power Output: 3.5 mm, 2-position terminal block.
- I/O Configuration
  - GPIOs: 4 GPIOs (I2C, UART, SPI compatible).
  - Power Pins: 1x 3.3VDC and 1x GND.
- Microcontroller
  - Model: ESP32-C3FH4 with 40 MHz crystal.
  - Wi-Fi: 802.11b/g/n.
  - Bluetooth: BLE 4.2.
  - Flash Memory: 4 MB.
- Interface
  - 3x LED Indicators
    - Power On
    - Output Enable
    - Programmable LED/Debug
  - Buttons
    - Reset
    - Programmable Button/Debug
- Programming
  - USB-C: Built-in USB JTAG programmer (ESP32-C3), compatible with Arduino. Select ESP32-C3 Dev Board.
  - TAG-Connect TC2050: Alternative programming option.
- Output
  - Current Sensor (CC6904SO-10A): Hall Effect Current Sensor.
  - Output Enable: DMP3017SFG-7 FET.

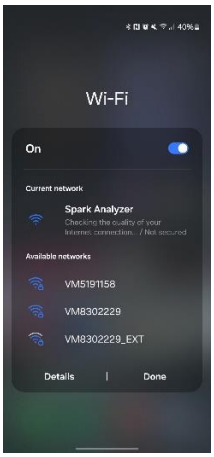
QUICKSTART | USING THE WEB APP OVER WIFI

1. Initial Setup

- Power On: Activate your Spark Analyzer. A blinking debug LED (5 times) indicates the firmware is correctly loaded.

2. WiFi Connection

- Connect to "Spark Analyzer" WiFi: Use a WiFi-enabled device like your phone or laptop to connect to the "Spark Analyzer" network. No internet is needed for this step.

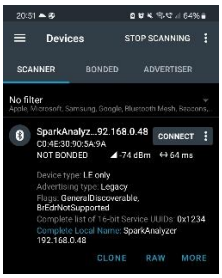


- Configure WiFi: Open a browser, go to 192.168.4.1 to access WiFiManager, and set up your home or office WiFi credentials.

3. Accessing the Web App

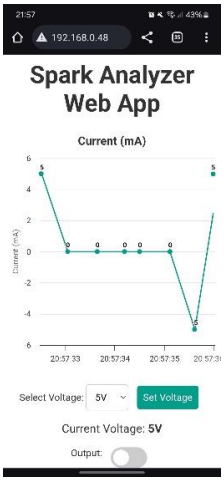
Find Device IP:

- Serial Monitor: With the device connected to USB, the IP address will appear in your IDE's serial monitor upon WiFi connection.
- BLE Advertisement: Use a BLE scanner app like NRF Connect to find the device broadcasting its IP address.



4. Web App Interface

- Launch Web App: Enter the IP address in your browser to open the Web App.



- Current Graph: Visualize real-time current measurements.
- Select and Set Voltage: Choose and apply the desired voltage.
- Output Control: Toggle the device's power output.

Troubleshooting

For WiFi or Web App issues, check WiFi range, credentials, and the serial monitor for errors.

TABLE OF CONTENTS

INTRODUCTION

The Spark Analyzer is an advanced tool that combines the functions of a USB-C Power Delivery Analyzer and a Programmable Power Supply. Featuring an ESP32 microcontroller, it is designed for precision in power management and analysis and is equipped with Wi-Fi and BLE for wireless control capabilities. This device is essential for a variety of applications, offering versatility in power delivery and monitoring.

KEY FEATURES

- **Low Noise Power Supply |** The Spark Analyzer is equipped with features for precise power control and analysis, enabling users to control the voltage and current via USB-C PD & PPS and measure the consumption using an on-board current sensor.
- **Programmable and Open Sourced |** Spark Analyzer is open source, enabling extensive customization and development. It is compatible with most IDEs including Arduino and PlatformIO.
- **Wireless Connectivity |** The Spark Analyzer boasts advanced wireless connectivity options, including both Wi-Fi and Bluetooth Low Energy (BLE), for improved flexibility, functionality, and user experience.
- **Compact Design |** Its compact form factor and user-friendly interface make the Spark Analyzer a convenient tool to integrate into your existing projects & prototypes with minimal redesign to give it full USB-C PD/PPS capability.

APPLICATION EXAMPLES

There are several example applications built for **Arduino IDE** & **PlatformIO.**

- **Prototyping with Programmable Power Supply** | The Spark Analyzer is ideal for home prototyping, providing a reliable and programmable power supply. This feature allows developers to test and power their projects with a wide range of voltage and current settings, ensuring flexibility and precision during the development process.
- **Project Flexibility for Higher Power Applications** | For projects requiring higher power, the Spark Analyzer ensures safe and flexible power delivery. Its ability to control and monitor power enables developers to safely explore and implement high-power applications, expanding the scope of their projects.
- **Education and Research** | In educational and research settings, the Spark Analyzer serves as a valuable tool for teaching and experimenting with power management concepts. Its advanced features and user-friendly interface make it suitable for students and researchers exploring the intricacies of power delivery and electronics design.

RELATED BOARDS

The Spark Analyzer’s capabilities make it candidate replacement for applications that might require the following boards:

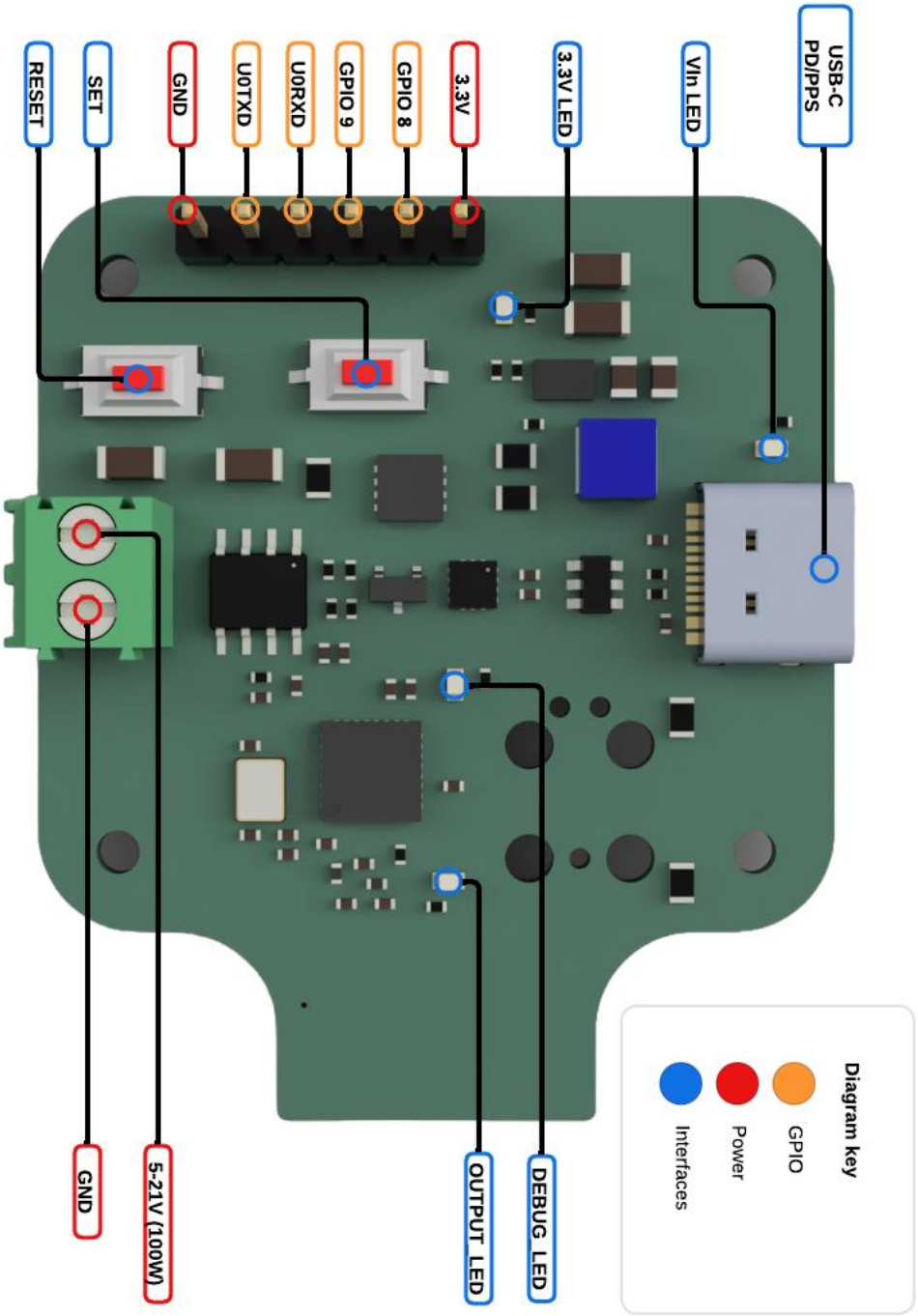
- PD Micro
- ESP32 Development Boards

OVERVIEW

RATINGS

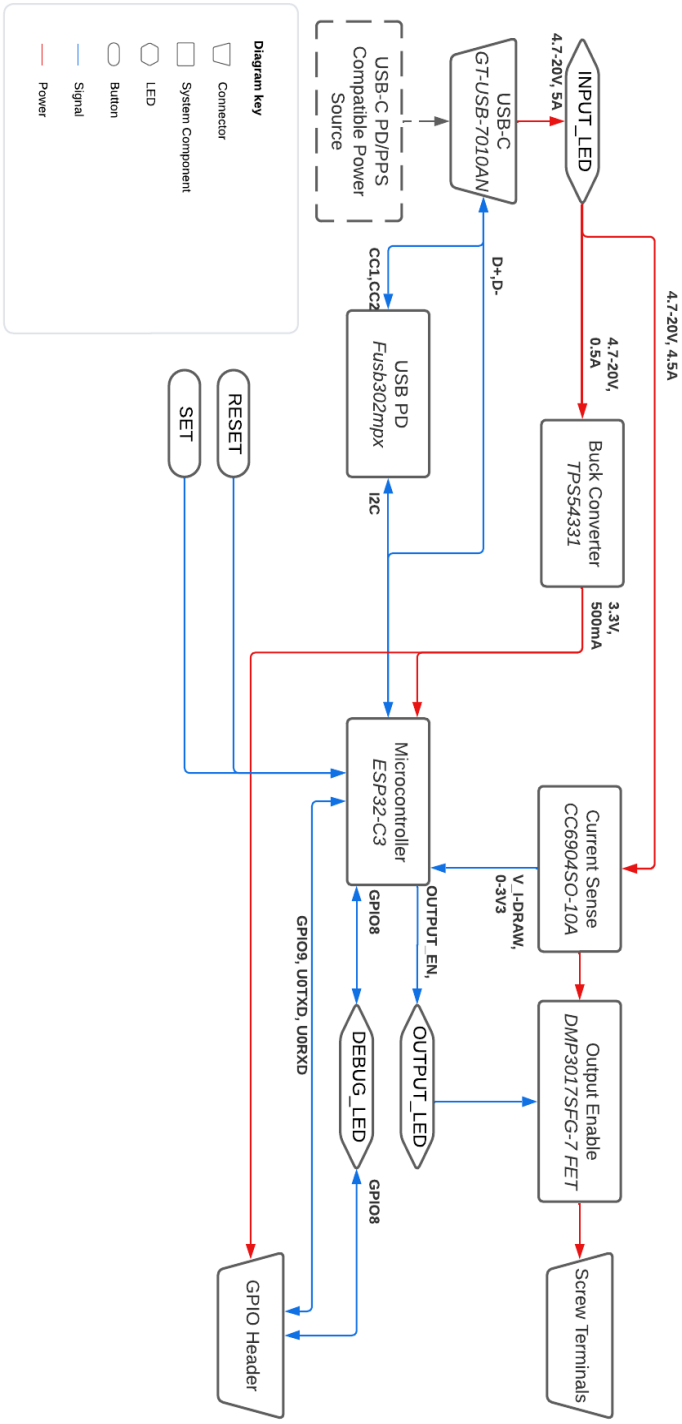
Symbol	Description	Min	Typ	Max	Unit
VIn	Input voltage range	3.3	5	21	V
VOut	Output voltage range	0	-	21	V
IOut	Max Output Current	-	-	5	A
PMax	Max Power			100	W
-	Thermal limits for board	-40 (-40)	-	85 (185)	°C (°F)

PINOUT DIAGRAM



Spark Analyzer Pinout Diagram

SYSTEM DIAGRAM



Spark Analyzer System Diagram

GETTING STARTED

USING ARDUINO IDE

This guide will assist you in setting up and utilizing the Spark Analyzer with the Arduino IDE, a user-friendly platform suitable for beginners and hobbyists. You'll learn how to load and run the Spark Analyzer's template firmware, which is a starting point for developing custom applications.

Prerequisites

- Spark Analyzer: Available on Crowdsupply & Mouser.
- Arduino IDE: Install the latest version of the Arduino IDE from the official Arduino website.
- USB-C Cable: For connecting the Spark Analyzer to your computer.
- USB PD/PPS source: e.g. Anker 511 Charger.

Step 1: Installing the ESP32 Board in Arduino IDE

1. Launch Arduino IDE.
2. Navigate to File > Preferences.

3. In the "Additional Board Manager URLs" field, enter: *https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json*
4. Click "OK" to save and exit Preferences.
5. Go to Tools > Board > Boards Manager.
6. Search for "ESP32", then install "ESP32 by Espressif Systems".

Step 2: Downloading Spark Analyzer Repository

1. Visit the Spark Analyzer GitHub repository - *https://github.com/tooyipjee/Spark-Analyzer/*
2. Download the repository ZIP file.
3. Extract the ZIP file to a convenient location on your computer.
4. Import the drivers needed to your IDE. They are in a folder named *src*

Step 3: Opening the Template Firmware Example

1. In the extracted folder, find the examples directory.
2. Locate the *Template* folder.
3. Open its .ino file with Arduino IDE.

Step 4: Uploading the Firmware

1. Connect the Spark Analyzer to your computer using a USB-C cable.
2. In Arduino IDE, select Tools > Board and choose the appropriate ESP32 board (e.g., ESP32 Dev Module).
3. Choose the right COM port under Tools > Port.
4. Under Tools, make sure.
    a. USB CDC On Boot is set to “Enabled.”
    b. JTAG Adapter is set to “USB Integrated JTAG.”
5. Press the "Upload" button in Arduino IDE to start uploading the firmware to your Spark Analyzer.

Step 5: Interacting with the Template Firmware

1. After uploading, the Spark Analyzer is set up with basic firmware, which includes:
    a. USB-C PD voltage – unplug it from your PC and connect to a PD compatible source.
    b. Real-time current measurement with a moving average filter.
    c. Configurable initial output state and current limit.
    d. Serial communication for status updates and debugging.

This template serves as a foundational base, allowing you to customize and expand upon it for your specific project needs.

Step 6: Experimenting with Other Examples

You can also explore other example projects. See Examples section.

Repeat Steps 3 and 4 to load and test these examples.

Arduino Examples

These examples provide a practical foundation for understanding how the Spark Analyzer can be integrated into different applications, ranging from remote control interfaces to smart home connectivity.

Bluetooth Controller Example

File: bluetooth_controller.ino

This example showcases the use of a graphical user interface (GUI) for the Spark Analyzer, focusing on smart home device power monitoring and control. It utilizes the RemoteXY library, enabling the creation of a GUI that can be controlled remotely through the RemoteXY mobile app. This setup is ideal for users looking to manage and monitor power delivery in a visually intuitive and user-friendly manner.

Flutter Controller Example

File: flutter_controller.ino

Designed for integration with a dedicated mobile app, this firmware emphasizes on BLE (Bluetooth Low Energy) communication and precise USB-C Power Delivery (PD) control. Key features include the ability to remotely control and monitor power settings, implement a moving average filter for stable current readings, and store user-defined voltage settings in EEPROM. This example is perfect for users who require remote power management with the convenience of a mobile app interface.

Matter Controller Example

File: matter_controller.ino

Focusing on smart home connectivity, this firmware integrates the Spark Analyzer with Matter (formerly Project CHIP). It demonstrates controlling an LED using Matter's OnOff cluster, along with integrating USB-C PD for effective voltage management. This example is particularly suitable for users interested in incorporating the Spark Analyzer into smart home systems, leveraging the advanced features of Matter for device control.

PPS (Programmable Power Supply) Example

File: pps.ino

This script is centered around the management of USB-C Power Delivery using the PD_UFP library. It demonstrates initializing and managing power delivery with specific voltage and current settings, using the FUSB302 chip for PD communication. This example is essential for users who need to control power delivery parameters through USB-C, offering a hands-on experience in managing power protocols.

Template Firmware Example

File: template.ino

Acting as a foundational base, this template firmware is designed for developers creating custom applications. It offers essential code for USB-C Power Delivery control and current monitoring. The template is easily adaptable and extendable, making it a valuable starting point for any project that requires precise voltage regulation and monitoring features.

USING PLATFORMIO

This guide will show you how to set up and use the Spark Analyzer with PlatformIO, an advanced and versatile development environment. The focus will be on the "Template" project from the Spark Analyzer's repository, which is designed for USB-C PD control and current monitoring.

Prerequisites

- Spark Analyzer: Available on Crowdsupply & Mouser.
- PlatformIO: Ensure PlatformIO is installed. You can use it either as a standalone IDE or as an extension for Visual Studio Code. Download from PlatformIO's website.
- USB-C Cable: For connecting the Spark Analyzer to your computer.
- USB PD/PPS source: e.g. Anker 511 Charger.

Step 1: Installing PlatformIO

1. If you choose to use PlatformIO with Visual Studio Code (VS Code), first install VS Code from the official site.
2. Open VS Code and go to the Extensions view by clicking on the square icon on the sidebar.
3. Search for "PlatformIO IDE" and install it.

Step 2: Downloading the Spark Analyzer Repository

1. Visit the Spark Analyzer GitHub repository - *https://github.com/tooyipjee/Spark-Analyzer/*
2. Download the repository as a ZIP file.
3. Extract the ZIP file to a known location on your computer.

Step 3: Opening the Template Project in PlatformIO

1. Launch PlatformIO from within VS Code.
2. Go to the "Home" screen and select "Open Project".
3. Browse to the location where you extracted the Spark Analyzer repository.
4. Navigate to the PlatformIO folder and then to the Template project folder.
5. Open this folder as a project in PlatformIO.

Step 4: Configuring the Project

Once the project is open, PlatformIO will automatically try to resolve and install any dependencies.

Step 5: Uploading the Firmware

1. Connect the Spark Analyzer to your computer via the USB-C cable.
2. Click on the "Upload" button in the PlatformIO toolbar (looks like a right arrow) to compile and upload the firmware.
3. The console at the bottom of the VS Code window will show the upload progress.

Step 6: Interacting with the Firmware

1. After uploading, the Spark Analyzer is set up with basic firmware, which includes:
    a. USB-C PD voltage – unplug it from your PC and connect to a PD compatible source.
    b. Real-time current measurement with a moving average filter.
    c. Configurable initial output state and current limit.
    d. Serial communication for status updates and debugging.

This template serves as a foundational base, allowing you to customize and expand upon it for your specific project needs.

PlatformIO Examples

These examples are essential for users looking to utilize the Spark Analyzer's full potential in varied applications, from basic functionality to advanced power management.

Template Firmware

The Template firmware for PlatformIO is an essential starting point for developers. It focuses on USB-C Power Delivery (PD) and current monitoring, making it ideal for projects requiring precise voltage control and real-time current measurements. The firmware includes function prototypes for better

code organization, configurable settings for monitoring intervals, and serial debugging support, making it highly adaptable for custom applications.

Template_PPS Firmware

This firmware specifically caters to the Programmable Power Supply (PPS) aspect of USB-C PD. It enables precise control over the power delivery, initializing and managing USB-C PD for specific voltage and current parameters. This script is particularly useful for applications where dynamic power supply adjustments are crucial, such as in smart charging stations or electronic testing equipment.

WebApp Firmware

The WebApp firmware provides a sophisticated user interface for controlling and monitoring power delivery, suitable for IoT and smart home applications. It uses SPIFFS for file storage, ESPAsyncWebServer for handling web requests, and WiFiManager for easy WiFi connectivity. This firmware is perfect for users who need a remote interface for real-time current monitoring, voltage adjustments, and power output control.

WebApp_PPS Firmware

Expanding on the WebApp firmware, the WebApp_PPS version integrates PPS features, offering advanced remote control over the Spark Analyzer's power delivery parameters. It's designed for applications that require detailed remote monitoring and control of voltage and current, such as in advanced home automation systems or remote laboratory setups.

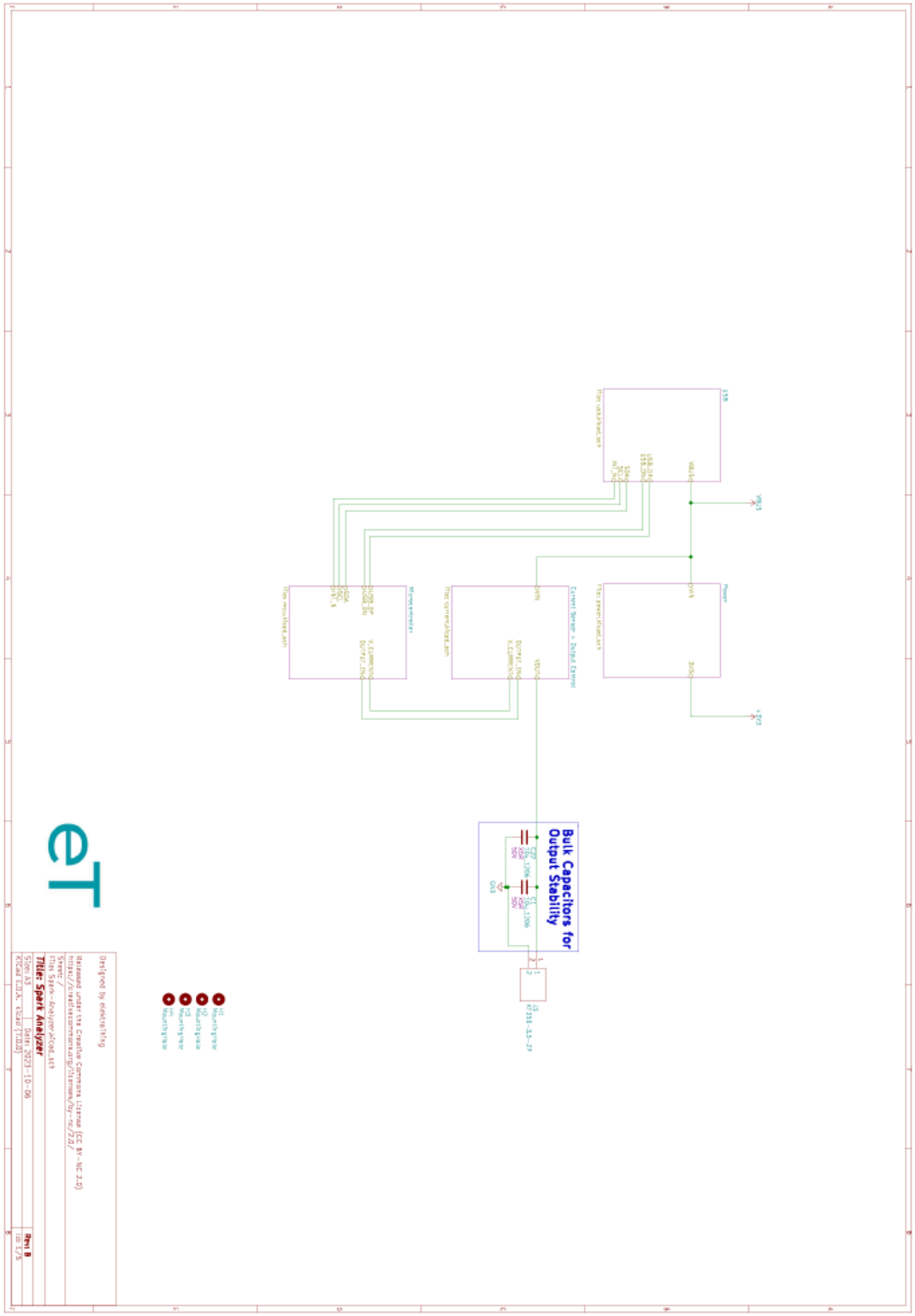
REFERENCES

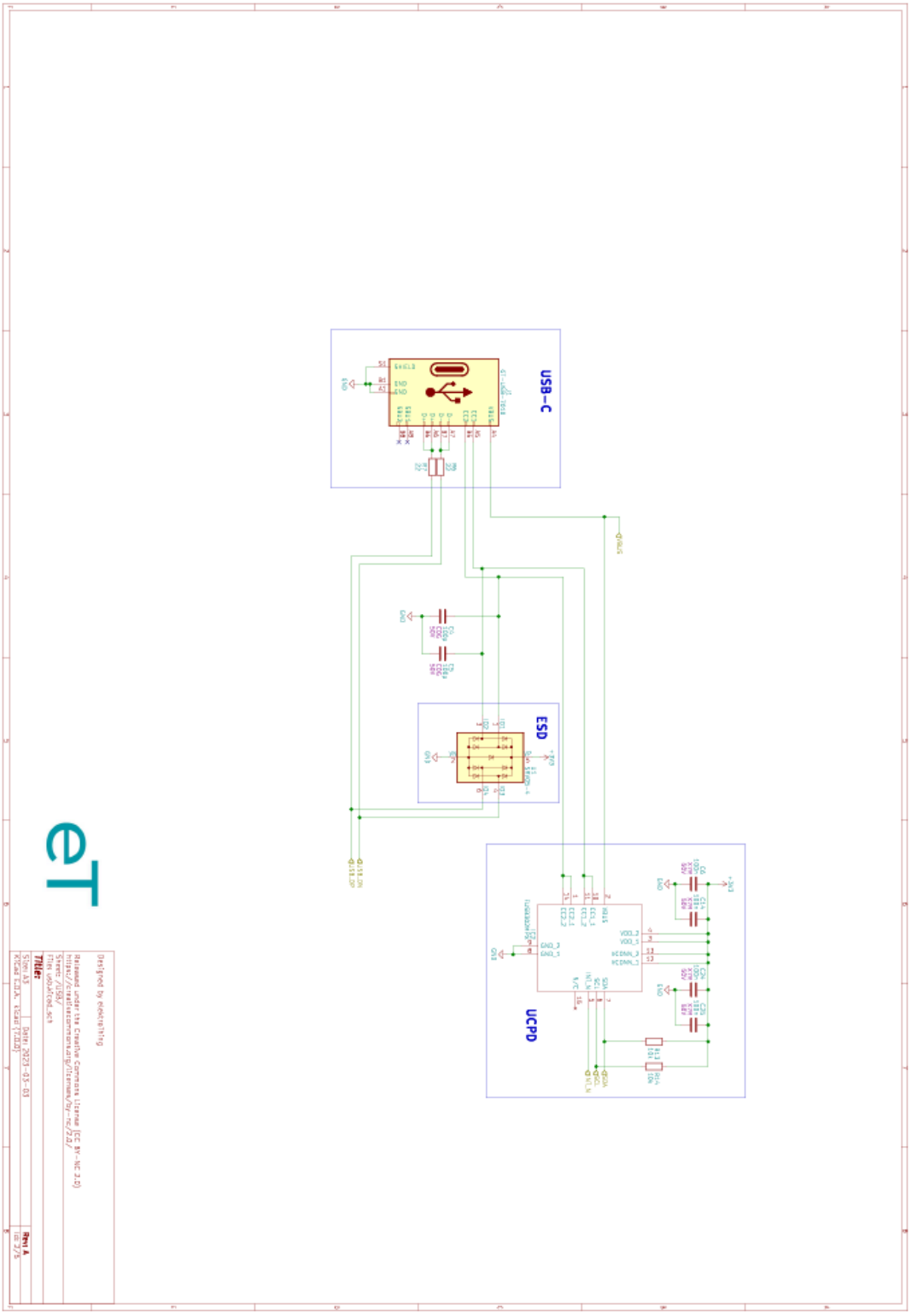
LINKS

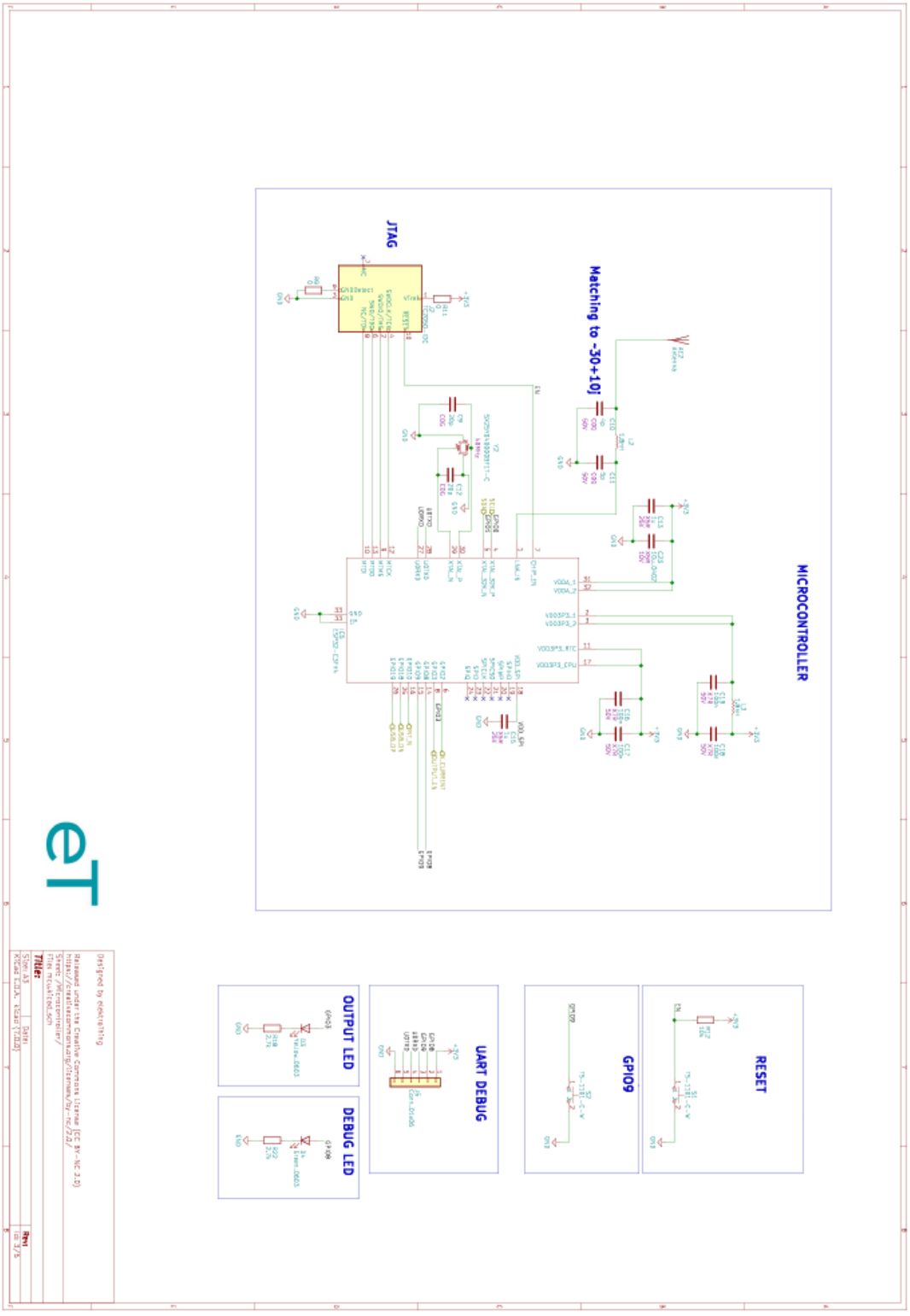
Here is a table of links to the key reference documents related to this product.

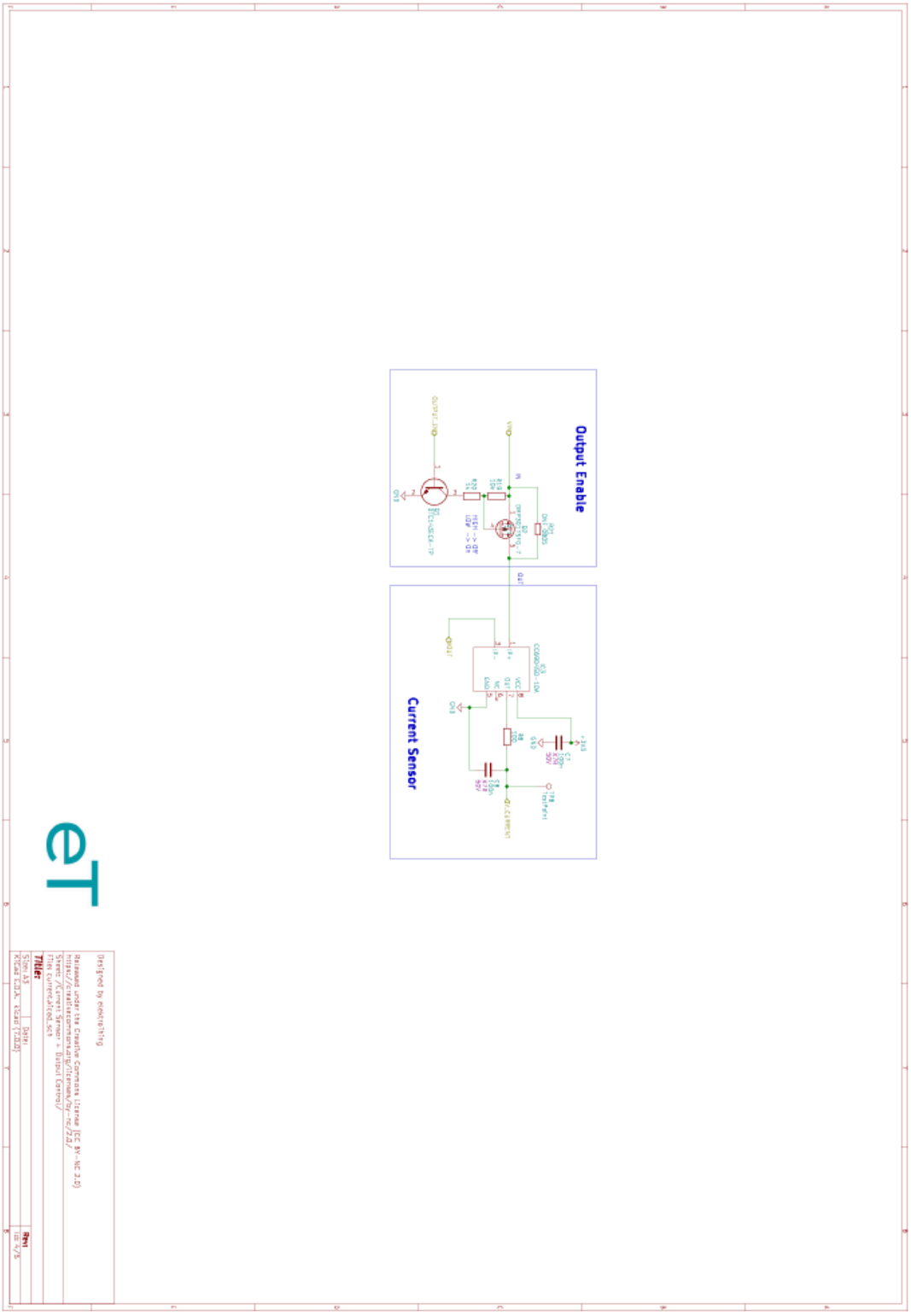
Reference	Link
Github Repository	*https://github.com/tooyipjee/Spark-Analyzer/*
Crowdsupply Campaign	*https://www.crowdsupply.com/elektrothing/spark-analyzer*
Website	*http://elektrothing.com/*

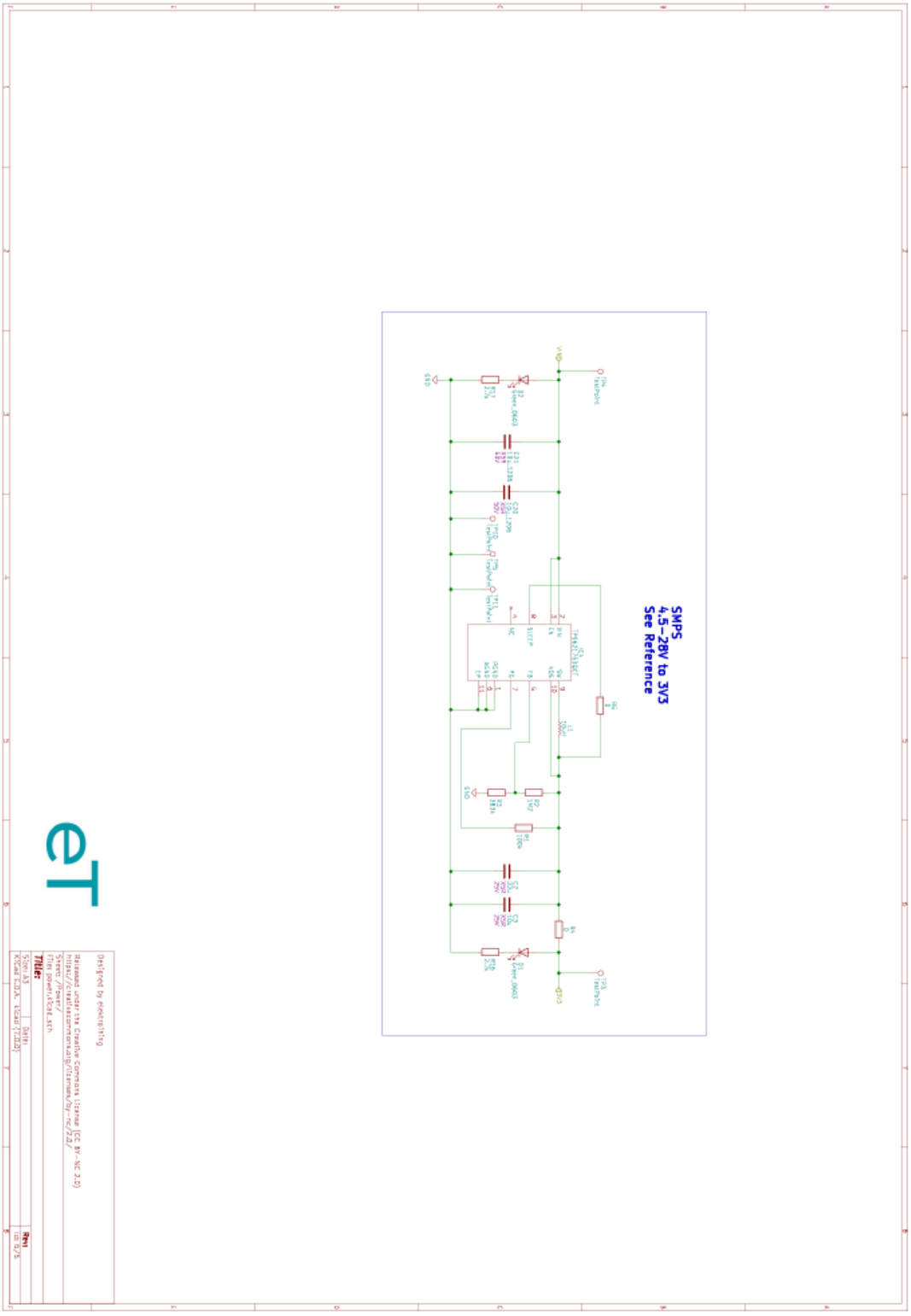
SCHEMATIC


Bulk Capacitors for
Output Stability
Title: Spark Analyzer

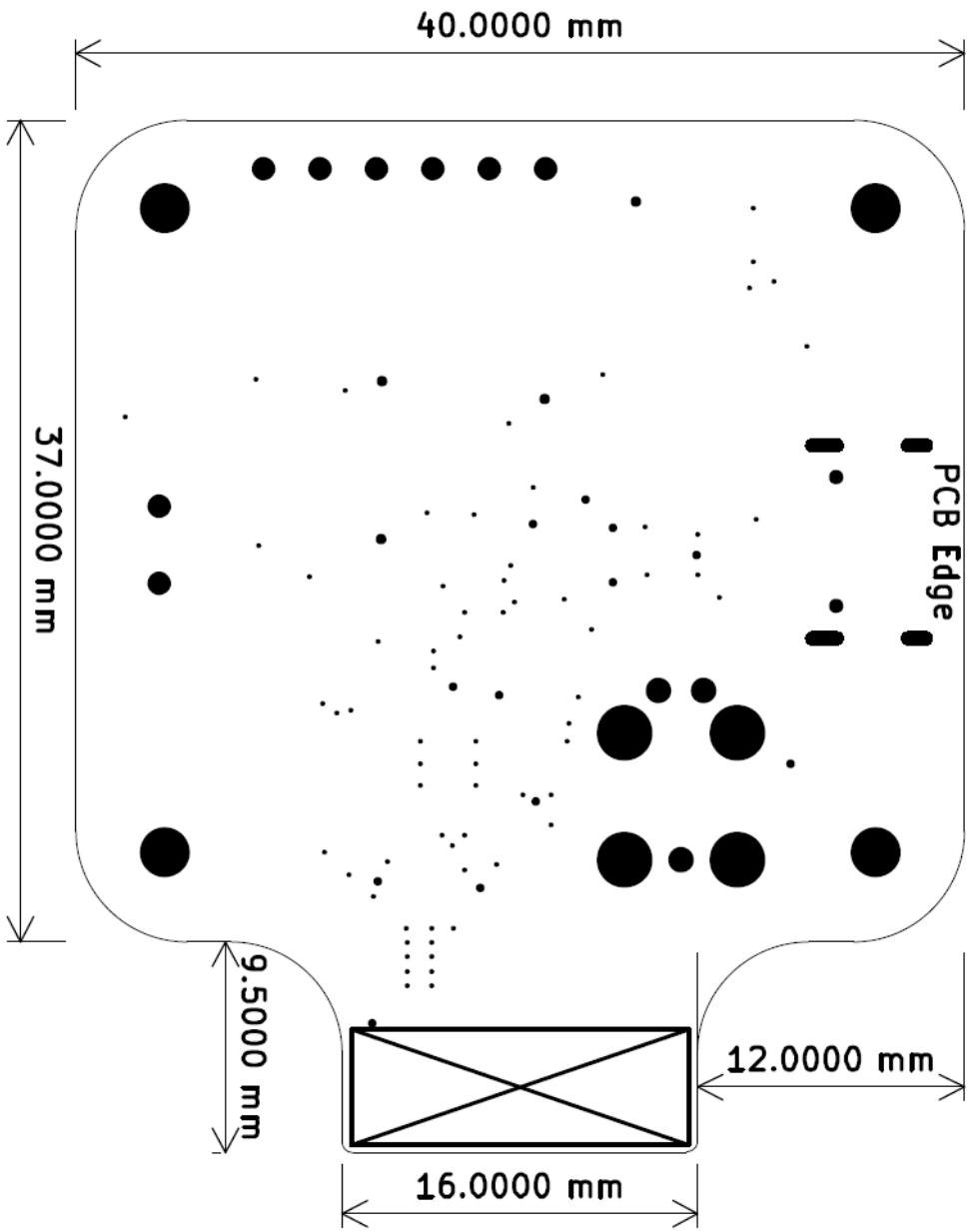

USB-C
ESD
UCPD


MICROCONTROLLER
Matching to ~30+10j
JTAG
RESET
GPIO9
UART DEBUG
OUTPUT LED
DEBUG LED


Output Enable
Current Sensor


SMPS
4.5–28V to 3V3
See Reference

PCB DIMENSIONS



Dimensions for Spark Analyzer

*Mounting holes are M2, 2.2mm.

CONFORMITY

The above reference product(s) is in conformity with the provisions of:

CE	2011/65/EU (ROHS) • IEC 62321-3-1:2013 • IEC 62321-5:2013 • IEC 62321-4:2013+AMD1: 2017 CSV • IEC 62321-6:2015 • IEC 62321-7-1:2015 • IEC 62321-7-2:2017 • IEC 62321-8:2017 2014/30/EU (EMC) • ETSI EN 301 489-1 V2.2.3:2019 • ETSI EN 301 489-17 V3.2.4:2020 • EN 55032:2015/A11:2020 • EN 55035:2017/A11:2020
UKCA	The Restriction of thex Use of Certain Hazardous Substances in Electrical and Electronic Equipment Regulations 2012, Statutory Instrument 2012/3032 (RoHS) • IEC 62321-3-1:2013 • IEC 62321-5:2013 • IEC 62321-4:2013+AMD1: 2017 CSV • IEC 62321-6:2015 • IEC 62321-7-1:2015 • IEC 62321-7-2:2017 • IEC 62321-8:2017 Electromagnetic Compatibility Regulations 2016 • ETSI EN 301 489-1 V2.2.3:2019 • ETSI EN 301 489-17 V3.2.4:2020