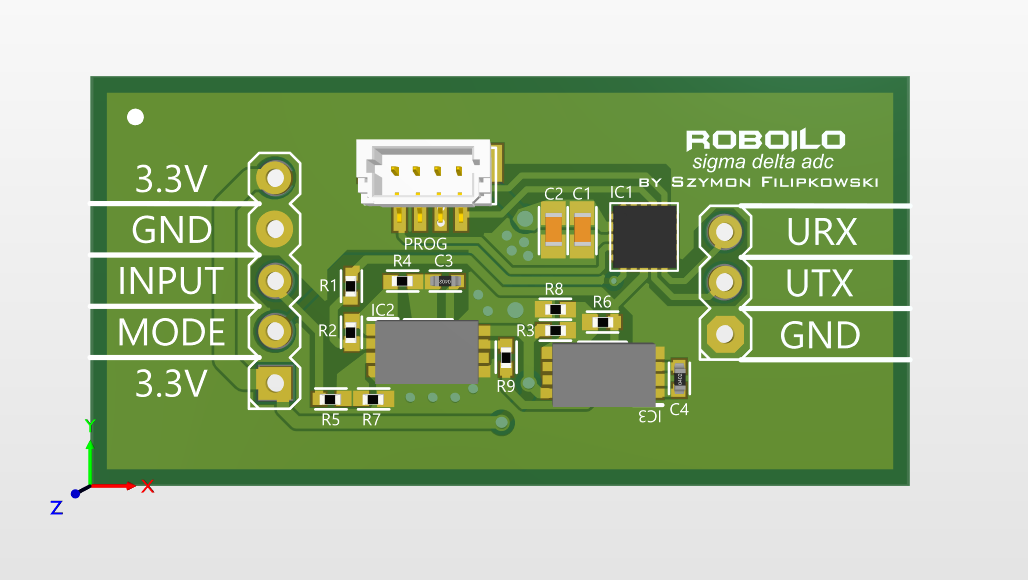
**SIGMA DELTA ADC MODULE**

**BY SZYMON FILIPKOWSKI**

Datasheet covers Sigma Delta ADC module by Szymon Filipkowski Rev 1.2.1  
Available in the near future on GITHUB: [“https://github.com/Tacot2009/Sigma-Delta-ADC-module-by-Szymon”.](https://github.com/Tacot2009/Sigma-Delta-ADC-module-by-Szymon-Filipkowski) **Power usage, sampling rate and bitrate are all approximation at this version.**

****

**Table of Contents**

1. Features
2. Applications
3. Description
4. Revision history
5. Pin Configuration and Functions
6. Specifications
7. Recommended Operating Conditions
8. AC Electrical Characteristic – single conversion mode
9. AC Electrical Characteristic – continuous conversion mode
10. Detailed Description
    1. Functional Block Diagram
    2. Analog Digital Approximation Mode Controll
    3. Operating Principle
11. **Features**

* On-Board MCU
* 0V to 2.56V analog input range
* Single 3.3V voltage supply
* UART one-way communication
* Easy mode change
* Plug-and-play feature
* Key specification
  + Resolution: 8bits
  + Accuracy at 0.01V
  + Sampling rate 2-10Hz
  + Power usage 2mW

1. **Applications**

* Operates with any device supporting UART communication protocol
* Interface to Temp Sensors, Voltage Sources, Transducers, Photoresistor,etc.

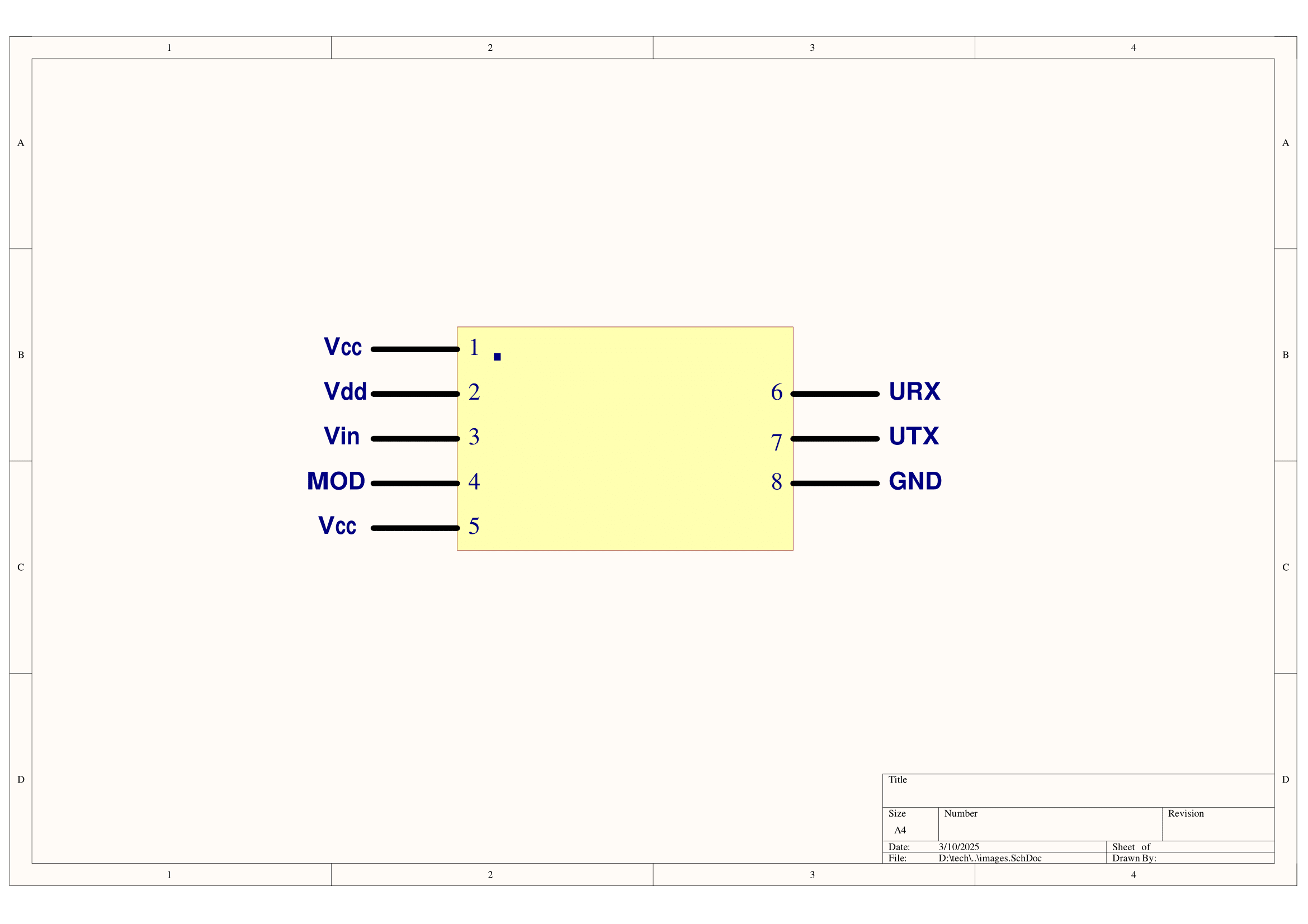
1. **Description**

“Sigma Delta ADC module by Szymon Filipkowski” is a 8bit successive approximation converters (ADC) that uses Sigma and Delta modulation to approximate digital value of input. This module uses stm32c0 mcu to decode digital signal into UART message and sends it to output device. Hardware and software is open sourced at GITHUB, so everyone can tailor this module to their specifics requirements. I as Szymon Filipkowski am creator and software developer for this project. I take full credit for this project.

1. **Revision history**

* Entry – first PCB project, software still in development. Datasheet created.
* v1.2.0 - added uart functionality in pcb, for future optional application in project
* v1.2.1 – translated this datasheet into polish

1. **Pin Configuration and Functions**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **PIN** | | **I/O** | **DESCRIPTION** |
| **NO.** | **NAME** |
| 1. | Vcc | I | +3.3V supply voltage |
| 2. | Vdd | I | Ground pin |
| 3. | Vin | I | Analog input |
| 4. | MOD | I | Mode select pin |
| 5. | Vcc | - | Vcc optional |
| 6. | URX | I | Uart receive data |
| 7. | UTX | O | Uart transmit data |
| 8. | GND | O | +0V reference for uart |

1. **Specifications**
   1. **Recommended Operating Conditions**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MIN** | **MAX** | **UNIT** |
| Vcc | 3.3 | 3.3 | V |
| Vdd | 0 | 0 | V |
| Analog Input Voltage | 0 | 2.56 | V |

* 1. **AC Electrical Characteristic – single conversion mode**

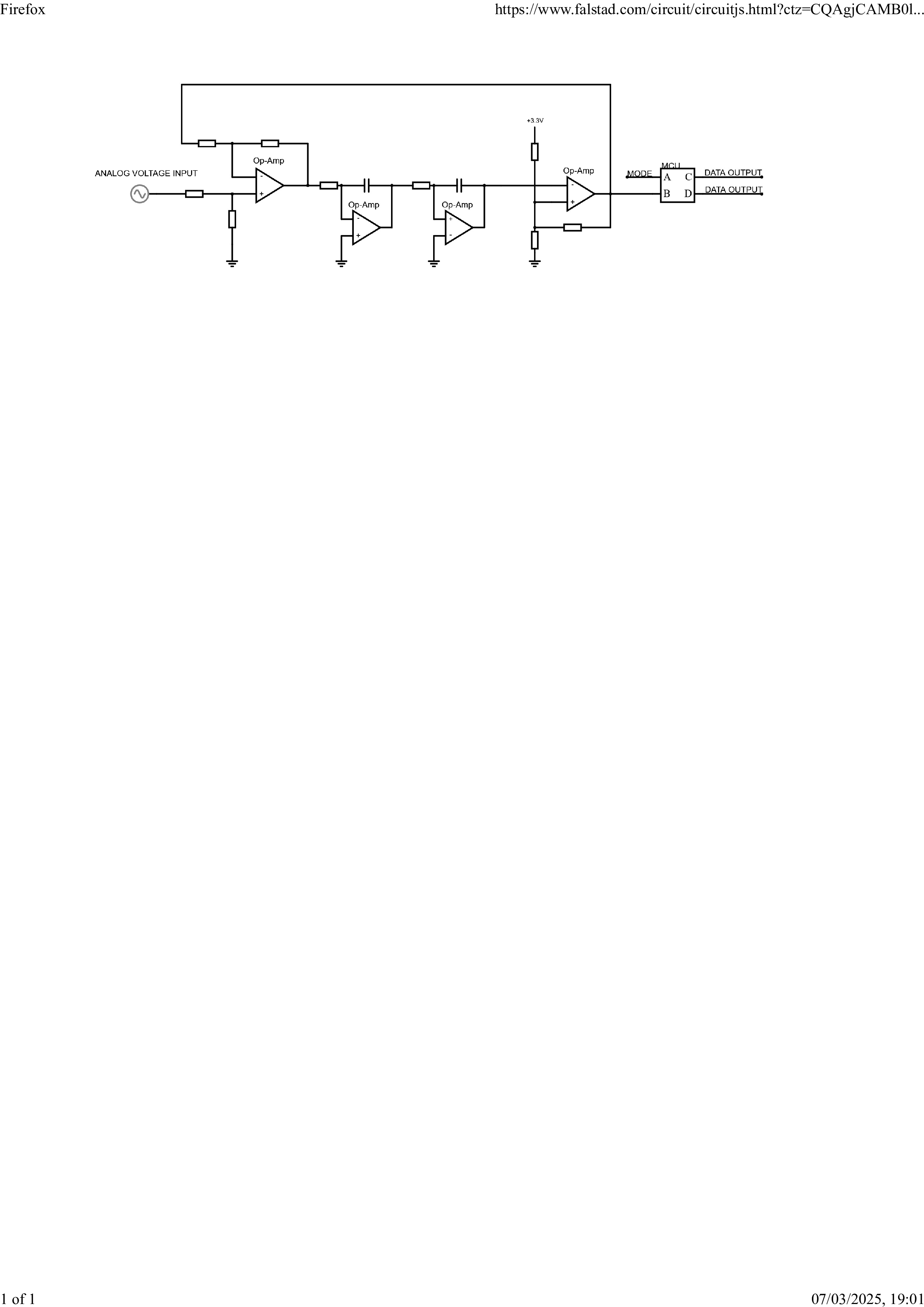
|  |  |  |
| --- | --- | --- |
|  | **TEST CONDITIONS** | **TYPICAL** |
| Single conversion time | Vin = 3.3V | 1 SEC |
| Power usage | Vin = 3.3V, f\*=1Hz | 2 mW |

\*f = speed of calling mcu to converse

* 1. **AC Electrical Characteristic – continuous conversion mode**

|  |  |  |
| --- | --- | --- |
|  | **TEST CONDITIONS** | **TYPICAL** |
| Sampling rate | Vin = 3.3V | 2Hz |
| Power usage | Vin = 3.3V | 2mW |

1. **Detailed Description**
   1. **Functional Block Diagram**



* 1. **Analog Digital Approximation Mode Controll**

This ADC device support two operating mode. Call mode (or single mode) and continuous mode (or auto mode). The auto mode is default for this module. Shorting MOD pin to 3.3V enables single mode. In this mode ADC makes one voltage approximation, sends it via UART protocol. After this MCU goes into STOP mode waiting for disconnecting MOD pin from 3.3V. MOD pin have pull-down. If MOD pin is left disconnected then module is in auto mode. In this mode module sends data via UART protocol as soon as calculation of input voltage completes.

* 1. **Operating Principle**
     + 1. The raw analog input is captured directly from source
       2. A operational amplifier configured in difference amplification subtracts the feedback voltage - delivered from the previous output - from the input signal. This step produces an error signal that shows the difference between the input and the current output.
       3. Two operational amplifiers, configured as integrators, process the error signal. These integrators filter out high-frequency noise and introduce a controlled delay. This integration improves signal accuracy and stabilizes the system dynamics prior to conversion.
       4. An operational amplifier, serving as a comparator, converts the integrated signal into a binary digital output (high or low). The resulting digital output is used as both the conversion result and as feedback for continuous error correction.