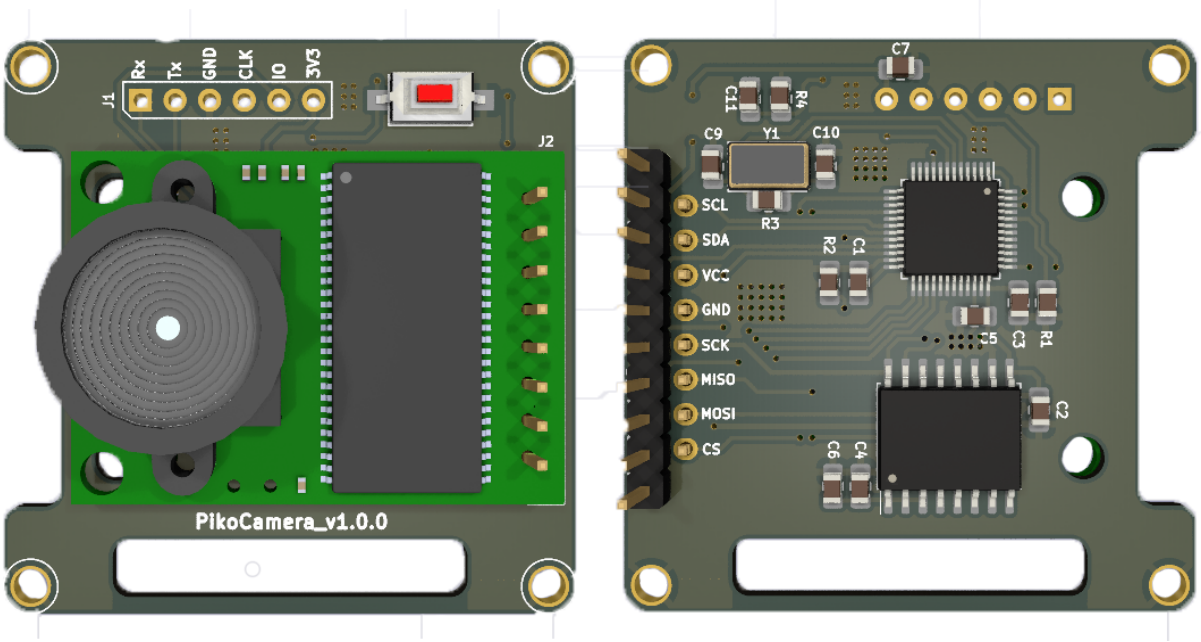
PikoCam Interface Control Document



May 2023
Issue: 1

Changes

Date	Changes	Pages/ Section	Responsible Person
May 2023	Document Creation		Rishav Mani Sharma, Jiten Thapa

Published by:
ORION Space Pvt. Ltd.
Kathmandu, Nepal
orionspace.nepal@gmail.com
www.orionspace.com.np

PikoCam

Introducing *PikoCam* - a cost-effective camera payload solution for picosatellites designed to provide reliable image transmission in harsh space environments. With its low power consumption, small form factor, and onboard implementation of the SSDV protocol, PikoCam makes imaging capability for pico/nano-satellites more accessible than ever.

PikoCam boasts a robust hardware design, utilizing COTS components throughout, including an STM32F103 microcontroller, 2MP ArduCam image sensor, and W25Q512NW flash memory. The module also features flexibility in its design, with additional pins for peripherals such as GPIO, SPI, and UART available on board. This allows for the integration of additional modules or the expansion of the module's capabilities.

Specifications

1. STM32F103: Cortex-M3 core, 1 Mbyte of Flash, 72 MHz CPU speed.
2. 64 megabytes of flash memory with an SPI interface for image storage.
3. Arducam with 2MP OV2648 sensor.
4. JPEG image format with SSDV protocol.
5. I2C interface with well-defined commands for the interface.
6. Extendable with extra accessible hardware peripherals.

SSDV is a robust digital packetized version of the JPEG image format for transmission over an unreliable medium. It has successfully been used in Lunar Satellite Missions and was authored by Philip Heron. (GitHub link: https://github.com/fsphil/ssdv).

Firmware updates and debug interface

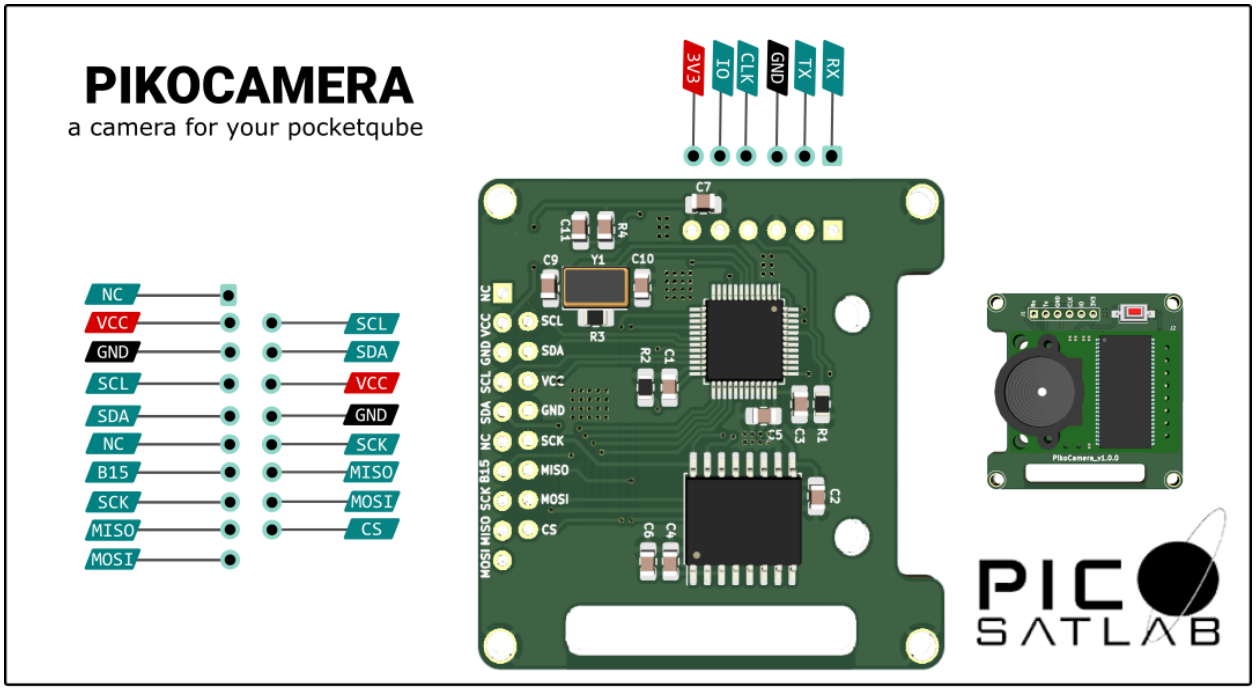


Fig. 1. Pin layout of PikoCam Board

PikoCam consists of an array with six pads consisting of power input, Serial Wire Debug (SWD), and UART. SWD is intended for firmware updates and UART to connect PikoCam to computers for testing. It can be used to communicate with other peripherals or modules too.

Rx	UART1 rx pin
Tx	UART1 tx pin
GND	0V wrt. 3V3
CLK	PA14 - SWD clock
IO	PA13 - SWD data
3V3	3.3V

Stack pin interface

PikoCam is designed such that it can be stacked on the customer's satellite boardswith other subsystems of PicoSatellite. There are two possible modes on which PikoCam can be used.

1. As an on-board computer: In this mode, other subsystems are connected to the PikoCam via stack pin interface and it handles mission goals as OBC as well as take images.
2. As standalone payload: In this mode, another OBC board on the picosatellite will command PikoCam to take images. In this mode, PikoCam works as a slave.

The stack pin interface consists of the following pins

MOSI	PB15 (SPI1)
MISO	PB14 (SPI1)
SCK	PB13 (SPI1)
B15	PB15 (SPIO)
SDA	PB11 (I2C2)
SCL	PB10 (I2C2)
VCC	3.3V
GND	0V wrt. VCC

There is the provision of SPI and I2C communication in the stack pin interface and UART in a six-pads array. This makes the board more versatile and adaptable to multiple systems. Sensors and modules with different communication protocols can be connected to PikoCam if required.

PikoCam Camera Test

In order to test the camera onboard PikoCam, flash `test.hex` from this link to the PikoCam.

If you are new to flashing hex files to STM32, you can follow the following steps:

1. Install ST-Link tools based on your operating system from https://github.com/stlink-org/stlink/releases/.
2. To check if the installation was successful, run the command `st-flash` in your command prompt.

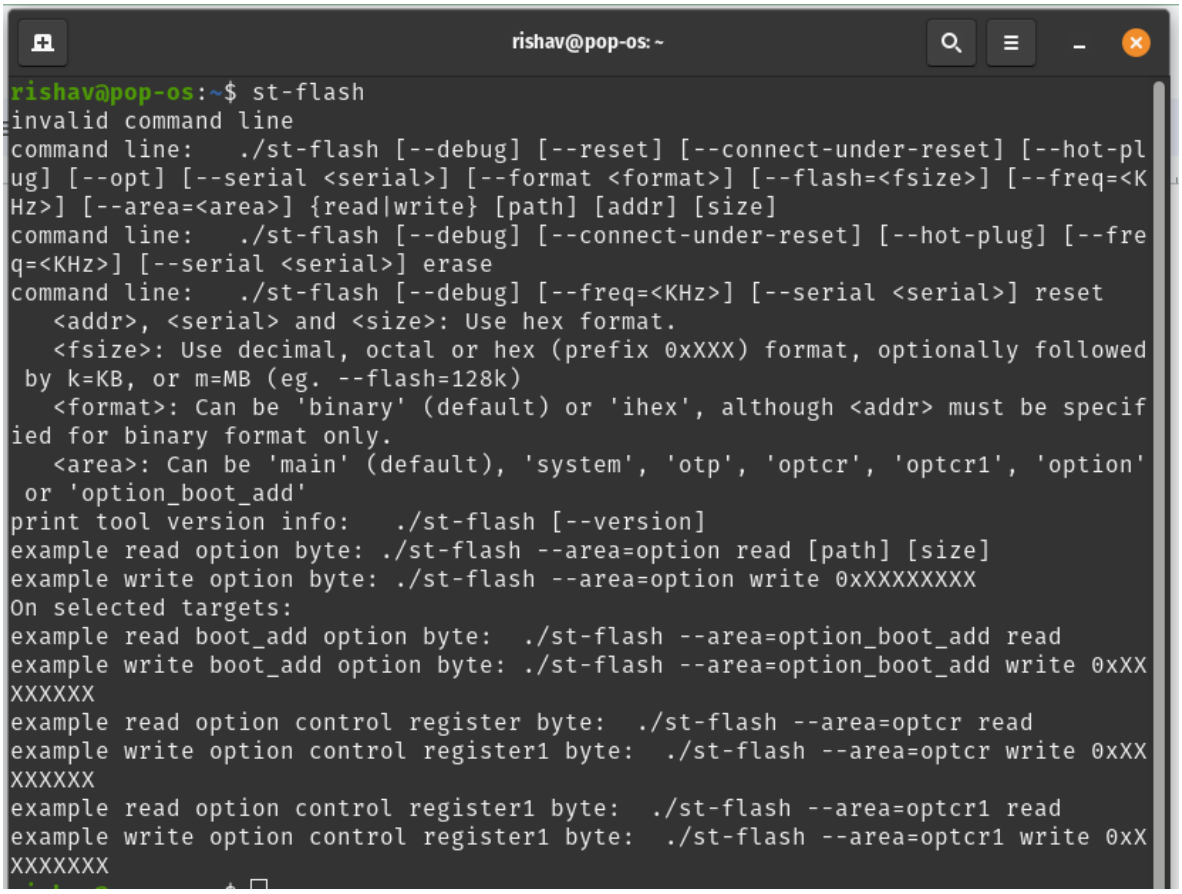


Fig 2. Successful message

3. Go inside the folder where the hex file resides.
4. Open the command prompt.
    a. For Windows: type cmd from the search bar of the folder you are inside.
    b. For Linux: Go inside the desired folder, right-click, and choose `Open in Terminal`.
5. Use the command
    ```
    st-flash write test.hex 0x8000000
    ```
    to flash the file named text.hex file to the PikoCam.
6. Connect PikoCam to the computer using UART1 pins via FTDI.

With the test firmware uploaded to the PikoCam, you can use this application called *ArduCAM Host* to capture images and view videos in real-time.

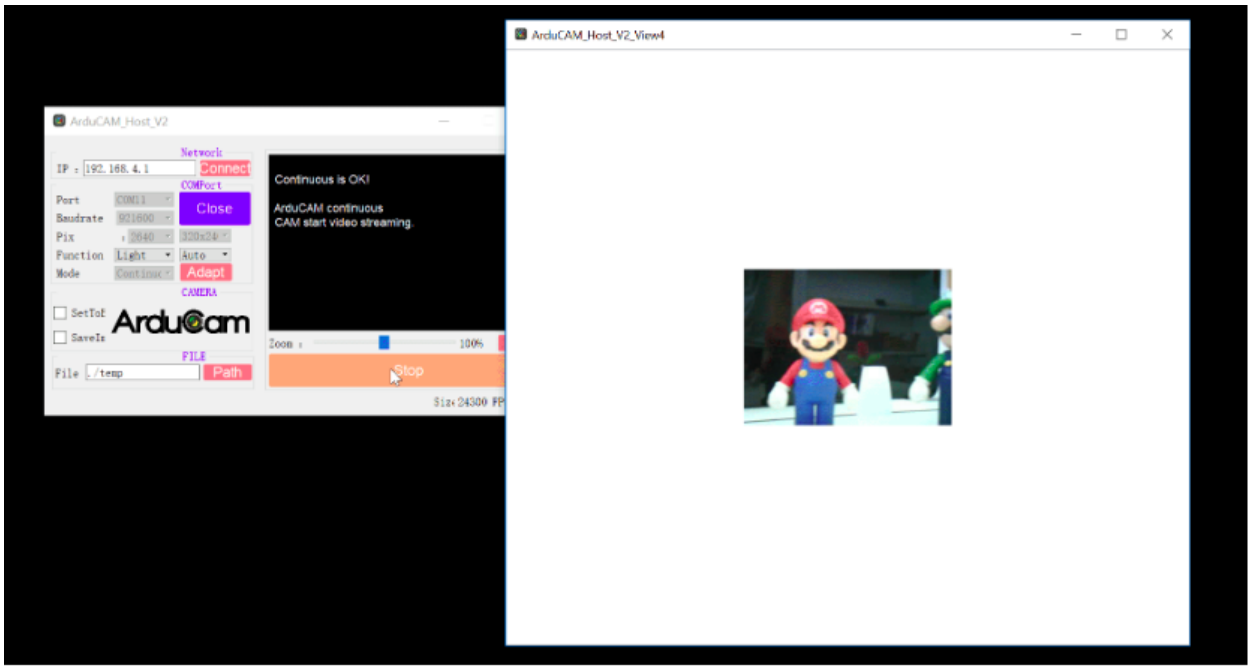


Fig. 3. Screenshot of *ArduCAM Host* in action

PikoCam SSDV test

Above test helps to verify that PikoCam is functioning and helps the user to tune the focus of the camera. This test is used to test the onboard SSDV encoder of PikoCam. In this test, PikoCam is used as a slave and an Arduino board as the master using the I2C protocol. Arduino commands PikoCam to take images and generate SSDV packets. The packets are sent to Arduino and are displayed on Serial Monitor as an array of hex. Finally, a CPP code will visualize the image corresponding to those SSDV packets.

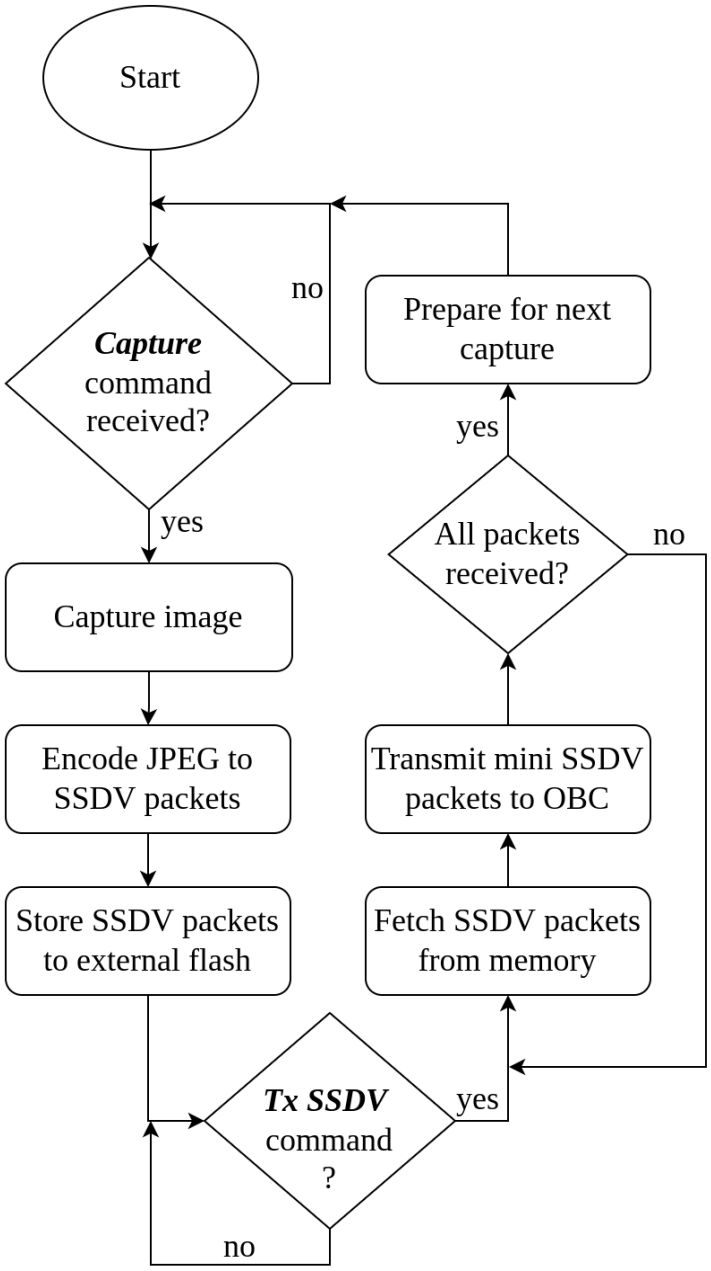
Following are the steps required for the PikoCam SSDV test. This video contains a demonstration of these steps.
(video link: https://www.youtube.com/watch?v=HCPtQSanssU&t=37s)

1. Connect Arduino with PikoCam with the following schematics

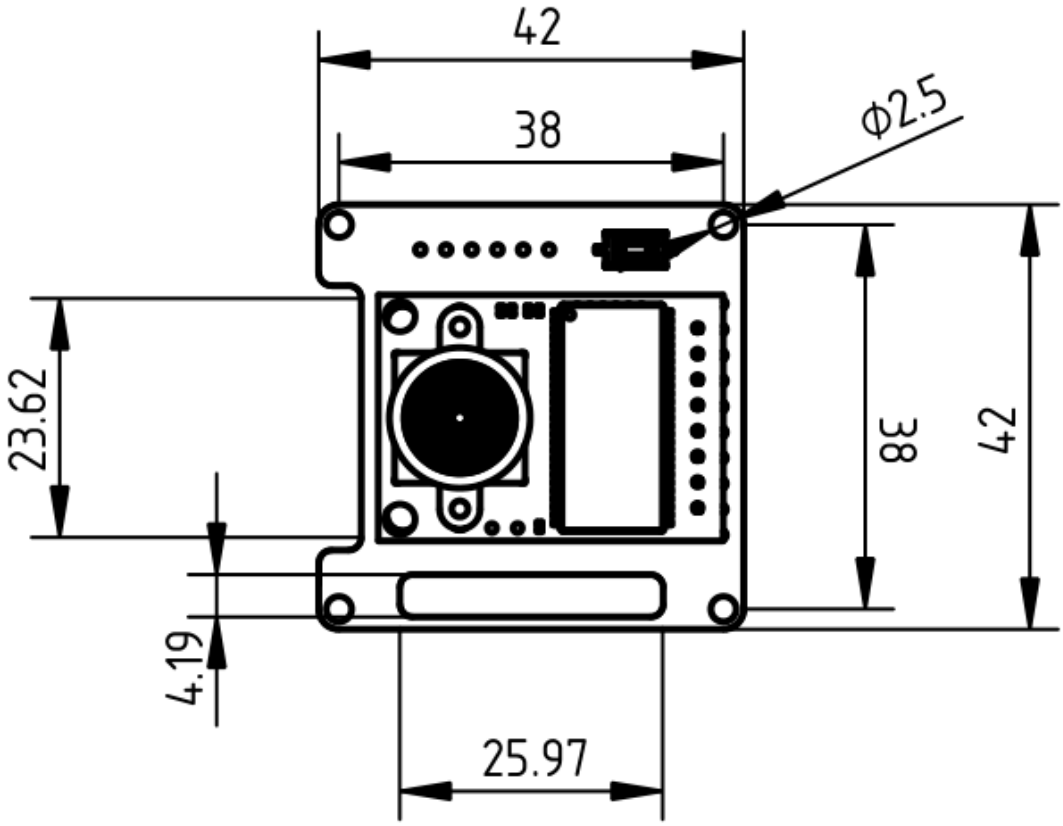
Arduino	PikoCam
3V3	VCC
GND	GND
SCL	SCL
SDA	SDA

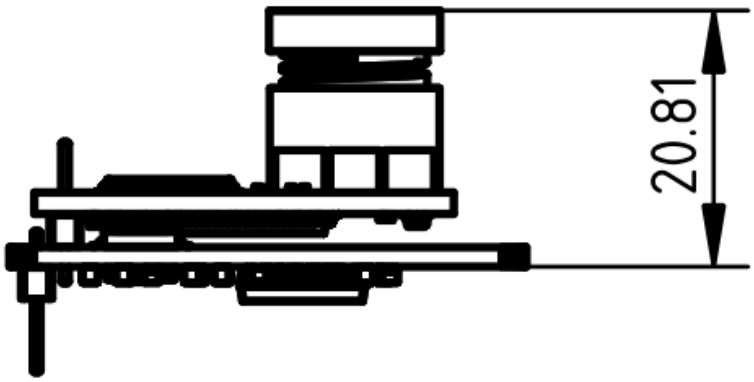
2. Open arduino_master on Arduino IDE and upload it to your Arduino.
3. Open Serial Monitor. The following prompt will appear on the monitor.
   /* Press 'a' to capture image */
4. Press ‘a’ and then press ‘enter’. The following message appears.
   /* Fetching SSDV packets. Please wait. */
5. The Arduino commands PikoCam to take an image, convert it to SSDV packets, and fetch them. The packets will appear on the monitor as elements of
   const uint8_t ssdv_packets[].
6. Copy the array and paste it to the header file ssdv_packets.h. It is located inside the folder ssdv_decoder.
7. Use the command make on the command prompt inside ssdv_decoder to run the code. The software will generate a jpeg image corresponding to the SSDV packets on the header ssdv_packets.h.

Software State Flow Diagram



Mechanical Drawing





All Dimensions in mm.