
Adafruit VC0706 Library Documentation

Release 1.0

Tony DiCola

Mar 02, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_vc0706	17
6.2.1	Implementation Notes	17
7	Indices and tables	19
	Python Module Index	21
	Index	23

CircuitPython module for use with the [VC0706 serial TTL camera](#). Allows basic image capture and download of image data from the camera over a serial connection. See examples for demo of saving image to a SD card (must be wired up separately) or internally.

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit library and driver bundle](#).

CHAPTER 2

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally [from PyPI](#). To install for current user:

```
pip3 install adafruit-circuitpython-vc0706
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-vc0706
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-vc0706
```


CHAPTER 3

Usage Example

See [examples/snapshot.py](#).

CHAPTER 4

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out [this guide](#).

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/vc0706_snapshot_filesystem.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """VC0706 image capture to local storage.
5  You must wire up the VC0706 to a USB or hardware serial port.
6  Primarily for use with Linux/Raspberry Pi but also can work with Mac/Windows"""
7
8  import time
9  import busio
10 import board
11 import adafruit_vc0706
12
13 # Set this to the full path to the file name to save the captured image. WILL_
14   ↳ OVERWRITE!
15 # CircuitPython internal filesystem configuration:
16 IMAGE_FILE = "/image.jpg"
17 # USB to serial adapter configuration:
18 # IMAGE_FILE = 'image.jpg' # Full path to file name to save captured image. Will_
19   ↳ overwrite!
20 # Raspberry Pi configuration:
21 # IMAGE_FILE = '/home/pi/image.jpg' # Full path to file name to save image. Will_
22   ↳ overwrite!
23
24 # Create a serial connection for the VC0706 connection.
25 uart = busio.UART(board.TX, board.RX, baudrate=115200, timeout=0.25)
26 # Update the serial port name to match the serial connection for the camera!
```

(continues on next page)

(continued from previous page)

```

25 # For use with USB to serial adapter:
26 # import serial
27 # uart = serial.Serial("/dev/ttyUSB0", baudrate=115200, timeout=0.25)
28 # For use with Raspberry Pi:
29 # import serial
30 # uart = serial.Serial("/dev/ttyS0", baudrate=115200, timeout=0.25)
31
32 # Setup VC0706 camera
33 vc0706 = adafruit_vc0706.VC0706(uart)
34
35 # Print the version string from the camera.
36 print("VC0706 version:")
37 print(vc0706.version)
38
39 # Set the image size.
40 vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480
41 # Or set IMAGE_SIZE_320x240 or IMAGE_SIZE_160x120
42
43 # Note you can also read the property and compare against those values to
44 # see the current size:
45 size = vc0706.image_size
46 if size == adafruit_vc0706.IMAGE_SIZE_640x480:
47     print("Using 640x480 size image.")
48 elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
49     print("Using 320x240 size image.")
50 elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
51     print("Using 160x120 size image.")
52
53 # Take a picture.
54 print("Taking a picture in 3 seconds...")
55 time.sleep(3)
56 print("SNAP!")
57 if not vc0706.take_picture():
58     raise RuntimeError("Failed to take picture!")
59
60 # Print size of picture in bytes.
61 frame_length = vc0706.frame_length
62 print("Picture size (bytes): {}".format(frame_length))
63
64 # Open a file for writing (overwriting it if necessary).
65 # This will write 50 bytes at a time using a small buffer.
66 # You MUST keep the buffer size under 100!
67 print("Writing image: {}".format(IMAGE_FILE), end="", flush=True)
68 stamp = time.monotonic()
69 # Pylint doesn't like the wcount variable being lowercase, but uppercase makes less_
   ↳ sense
70 # pylint: disable=invalid-name
71 with open(IMAGE_FILE, "wb") as outfile:
72     wcount = 0
73     while frame_length > 0:
74         t = time.monotonic()
75         # Compute how much data is left to read as the lesser of remaining bytes
76         # or the copy buffer size (32 bytes at a time). Buffer size MUST be
77         # a multiple of 4 and under 100. Stick with 32!
78         to_read = min(frame_length, 32)
79         copy_buffer = bytearray(to_read)
80         # Read picture data into the copy buffer.

```

(continues on next page)

(continued from previous page)

```

81     if vc0706.read_picture_into(copy_buffer) == 0:
82         raise RuntimeError("Failed to read picture frame data!")
83     # Write the data to SD card file and decrement remaining bytes.
84     outfile.write(copy_buffer)
85     frame_length -= 32
86     # Print a dot every 2k bytes to show progress.
87     wcount += 1
88     if wcount >= 64:
89         print(".", end="", flush=True)
90         wcount = 0
91 print()
92 # pylint: enable=invalid-name
93 print("Finished in %0.1f seconds!" % (time.monotonic() - stamp))
94 # Turn the camera back into video mode.
95 vc0706.resume_video()

```

Listing 2: examples/vc0706_snapshot_simpletest.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """VC0706 image capture to SD card demo.
5  You must wire up the VC0706 to the board's serial port, and a SD card holder
6  to the board's SPI bus. Use the Feather M0 Adalogger as it includes a SD
7  card holder pre-wired to the board--this sketch is setup to use the Adalogger!
8  In addition you MUST also install the following dependent SD card library:
9  https://github.com/adafruit/Adafruit_CircuitPython_SD
10 See the guide here for more details on using SD cards with CircuitPython:
11 https://learn.adafruit.com/micropython-hardware-sd-cards"""
12
13 import time
14
15 import board
16 import busio
17 import digitalio
18 import storage
19
20 import adafruit_sdcard
21 import adafruit_vc0706
22
23
24 # Configuration:
25 SD_CS_PIN = board.D10 # CS for SD card (SD_CS is for Feather Adalogger)
26 IMAGE_FILE = "/sd/image.jpg" # Full path to file name to save captured image.
27 # Will overwrite!
28
29 # Setup SPI bus (hardware SPI).
30 spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
31
32 # Setup SD card and mount it in the filesystem.
33 sd_cs = digitalio.DigitalInOut(SD_CS_PIN)
34 sdcard = adafruit_sdcard.SDCard(spi, sd_cs)
35 vfs = storage.VfsFat(sdcard)
36 storage.mount(vfs, "/sd")
37
38 # Create a serial connection for the VC0706 connection, speed is auto-detected.

```

(continues on next page)

(continued from previous page)

```

39 uart = busio.UART(board.TX, board.RX)
40 # Setup VC0706 camera
41 vc0706 = adafruit_vc0706.VC0706(uart)
42
43 # Print the version string from the camera.
44 print("VC0706 version:")
45 print(vc0706.version)
46
47 # Set the baud rate to 115200 for fastest transfer (its the max speed)
48 vc0706.baudrate = 115200
49
50 # Set the image size.
51 vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480 # Or set IMAGE_SIZE_320x240 or
52 # IMAGE_SIZE_160x120
53 # Note you can also read the property and compare against those values to
54 # see the current size:
55 size = vc0706.image_size
56 if size == adafruit_vc0706.IMAGE_SIZE_640x480:
57     print("Using 640x480 size image.")
58 elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
59     print("Using 320x240 size image.")
60 elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
61     print("Using 160x120 size image.")
62
63 # Take a picture.
64 print("Taking a picture in 3 seconds...")
65 time.sleep(3)
66 print("SNAP!")
67 if not vc0706.take_picture():
68     raise RuntimeError("Failed to take picture!")
69
70 # Print size of picture in bytes.
71 frame_length = vc0706.frame_length
72 print("Picture size (bytes): {}".format(frame_length))
73
74 # Open a file for writing (overwriting it if necessary).
75 # This will write 50 bytes at a time using a small buffer.
76 # You MUST keep the buffer size under 100!
77 print("Writing image: {}".format(IMAGE_FILE), end="")
78 stamp = time.monotonic()
79 # pylint: disable=invalid-name
80 with open(IMAGE_FILE, "wb") as outfile:
81     wcount = 0
82     while frame_length > 0:
83         # Compute how much data is left to read as the lesser of remaining bytes
84         # or the copy buffer size (32 bytes at a time). Buffer size MUST be
85         # a multiple of 4 and under 100. Stick with 32!
86         to_read = min(frame_length, 32)
87         copy_buffer = bytearray(to_read)
88         # Read picture data into the copy buffer.
89         if vc0706.read_picture_into(copy_buffer) == 0:
90             raise RuntimeError("Failed to read picture frame data!")
91         # Write the data to SD card file and decrement remaining bytes.
92         outfile.write(copy_buffer)
93         frame_length -= 32
94         # Print a dot every 2k bytes to show progress.
95         wcount += 1

```

(continues on next page)

(continued from previous page)

```

96         if wcount >= 64:
97             print(".", end="")
98             wcount = 0
99 # pylint: enable=invalid-name
100 print()
101 print("Finished in %0.1f seconds!" % (time.monotonic() - stamp))
102 # Turn the camera back into video mode.
103 vc0706.resume_video()

```

6.2 adafruit_vc0706

VC0706 serial TTL camera module. Allows basic image capture and download of image data from the camera over a serial connection. See examples for demo of saving image to a SD card (must be wired up separately).

- Author(s): Tony DiCola

6.2.1 Implementation Notes

Hardware:

- Adafruit [TTL Serial JPEG Camera with NTSC Video](#) (Product ID: 397)

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: <https://github.com/adafruit/circuitpython/releases>

class `adafruit_vc0706.VC0706(uart, *, buffer_size=100)`

Driver for VC0706 serial TTL camera module. :param ~busio.UART uart: uart serial or compatible interface
:param int buffer_size: Receive buffer size

baudrate

Return the currently configured baud rate.

frame_length

Return the length in bytes of the currently capture frame/picture.

image_size

Get the current image size, will return a value of IMAGE_SIZE_640x480, IMAGE_SIZE_320x240, or IMAGE_SIZE_160x120.

read_picture_into(buf)

Read the next bytes of frame/picture data into the provided buffer. Returns the number of bytes written to the buffer (might be less than the size of the buffer). Buffer MUST be a multiple of 4 and 100 or less. Suggested buffer size is 32.

resume_video()

Tell the camera to resume being a camera after the video has stopped (Such as what happens when a picture is taken).

take_picture()

Tell the camera to take a picture. Returns True if successful.

version

Return camera version byte string.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

a

adafruit_vc0706, [17](#)

A

`adafruit_vc0706` (*module*), [17](#)

B

`baudrate` (*adafruit_vc0706.VC0706 attribute*), [17](#)

F

`frame_length` (*adafruit_vc0706.VC0706 attribute*),
[17](#)

I

`image_size` (*adafruit_vc0706.VC0706 attribute*), [17](#)

R

`read_picture_into()` (*adafruit_vc0706.VC0706 method*), [17](#)

`resume_video()` (*adafruit_vc0706.VC0706 method*),
[17](#)

T

`take_picture()` (*adafruit_vc0706.VC0706 method*),
[17](#)

V

`VC0706` (*class in adafruit_vc0706*), [17](#)

`version` (*adafruit_vc0706.VC0706 attribute*), [17](#)