
Adafruit VC0706 Library Documentation

Release 1.0

Tony DiCola

Aug 25, 2020

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	16
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  """VC0706 image capture to local storage.
2  You must wire up the VC0706 to a USB or hardware serial port.
3  Primarily for use with Linux/Raspberry Pi but also can work with Mac/Windows"""
4
5  import time
6  import busio
7  import board
8  import adafruit_vc0706
9
10 # Set this to the full path to the file name to save the captured image. Will_
    ↳ OVERWRITE!
11 # CircuitPython internal filesystem configuration:
12 IMAGE_FILE = "/image.jpg"
13 # USB to serial adapter configuration:
14 # IMAGE_FILE = 'image.jpg' # Full path to file name to save captured image. Will_
    ↳ overwrite!
15 # Raspberry Pi configuration:
16 # IMAGE_FILE = '/home/pi/image.jpg' # Full path to file name to save image. Will_
    ↳ overwrite!
17
18
19 # Create a serial connection for the VC0706 connection.
20 uart = busio.UART(board.TX, board.RX, baudrate=115200, timeout=0.25)
21 # Update the serial port name to match the serial connection for the camera!
22 # For use with USB to serial adapter:
23 # import serial
24 # uart = serial.Serial("/dev/ttyUSB0", baudrate=115200, timeout=0.25)

```

(continues on next page)

(continued from previous page)

```

25 # For use with Raspberry Pi:
26 # import serial
27 # uart = serial.Serial("/dev/ttyS0", baudrate=115200, timeout=0.25)
28
29 # Setup VC0706 camera
30 vc0706 = adafruit_vc0706.VC0706(uart)
31
32 # Print the version string from the camera.
33 print("VC0706 version:")
34 print(vc0706.version)
35
36 # Set the image size.
37 vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480
38 # Or set IMAGE_SIZE_320x240 or IMAGE_SIZE_160x120
39
40 # Note you can also read the property and compare against those values to
41 # see the current size:
42 size = vc0706.image_size
43 if size == adafruit_vc0706.IMAGE_SIZE_640x480:
44     print("Using 640x480 size image.")
45 elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
46     print("Using 320x240 size image.")
47 elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
48     print("Using 160x120 size image.")
49
50 # Take a picture.
51 print("Taking a picture in 3 seconds...")
52 time.sleep(3)
53 print("SNAP!")
54 if not vc0706.take_picture():
55     raise RuntimeError("Failed to take picture!")
56
57 # Print size of picture in bytes.
58 frame_length = vc0706.frame_length
59 print("Picture size (bytes): {}".format(frame_length))
60
61 # Open a file for writing (overwriting it if necessary).
62 # This will write 50 bytes at a time using a small buffer.
63 # You MUST keep the buffer size under 100!
64 print("Writing image: {}".format(IMAGE_FILE), end="", flush=True)
65 stamp = time.monotonic()
66 with open(IMAGE_FILE, "wb") as outfile:
67     wcount = 0
68     while frame_length > 0:
69         t = time.monotonic()
70         # Compute how much data is left to read as the lesser of remaining bytes
71         # or the copy buffer size (32 bytes at a time). Buffer size MUST be
72         # a multiple of 4 and under 100. Stick with 32!
73         to_read = min(frame_length, 32)
74         copy_buffer = bytearray(to_read)
75         # Read picture data into the copy buffer.
76         if vc0706.read_picture_into(copy_buffer) == 0:
77             raise RuntimeError("Failed to read picture frame data!")
78         # Write the data to SD card file and decrement remaining bytes.
79         outfile.write(copy_buffer)
80         frame_length -= 32
81         # Print a dot every 2k bytes to show progress.

```

(continues on next page)

(continued from previous page)

```

82     wcount += 1
83     if wcount >= 64:
84         print(".", end="", flush=True)
85         wcount = 0
86 print()
87 print("Finished in %0.1f seconds!" % (time.monotonic() - stamp))

```

Listing 2: examples/vc0706_snapshot_simpletest.py

```

1  """VC0706 image capture to SD card demo.
2  You must wire up the VC0706 to the board's serial port, and a SD card holder
3  to the board's SPI bus. Use the Feather M0 Adalogger as it includes a SD
4  card holder pre-wired to the board--this sketch is setup to use the Adalogger!
5  In addition you MUST also install the following dependent SD card library:
6  https://github.com/adafruit/Adafruit_CircuitPython_SD
7  See the guide here for more details on using SD cards with CircuitPython:
8  https://learn.adafruit.com/micropython-hardware-sd-cards"""
9
10 import time
11
12 import board
13 import busio
14 import digitalio
15 import storage
16
17 import adafruit_sdcard
18 import adafruit_vc0706
19
20
21 # Configuration:
22 SD_CS_PIN = board.D10 # CS for SD card (SD_CS is for Feather Adalogger)
23 IMAGE_FILE = "/sd/image.jpg" # Full path to file name to save captured image.
24 # Will overwrite!
25
26 # Setup SPI bus (hardware SPI).
27 spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
28
29 # Setup SD card and mount it in the filesystem.
30 sd_cs = digitalio.DigitalInOut(SD_CS_PIN)
31 sdcard = adafruit_sdcard.SDCard(spi, sd_cs)
32 vfs = storage.VfsFat(sdcard)
33 storage.mount(vfs, "/sd")
34
35 # Create a serial connection for the VC0706 connection, speed is auto-detected.
36 uart = busio.UART(board.TX, board.RX, timeout=250)
37 # Setup VC0706 camera
38 vc0706 = adafruit_vc0706.VC0706(uart)
39
40 # Print the version string from the camera.
41 print("VC0706 version:")
42 print(vc0706.version)
43
44 # Set the baud rate to 115200 for fastest transfer (its the max speed)
45 vc0706.baudrate = 115200
46
47 # Set the image size.

```

(continues on next page)

(continued from previous page)

```

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

```

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.

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, [16](#)

A

`adafruit_vc0706` (*module*), [16](#)

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](#)

T

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

V

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

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