
Adafruit VC0706 Library Documentation

Release 1.0

Tony DiCola

Dec 21, 2018

Contents

1	Dependencies	3
2	Usage Example	5
3	Contributing	7
4	Building locally	9
4.1	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_vc0706	14
5.2.1	Implementation Notes	14
6	Indices and tables	17
	Python Module Index	19

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

Usage Example

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

CHAPTER 3

Contributing

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

CHAPTER 4

Building locally

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-vc0706 --library_
↪location .
```

4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/snapshot_internal.py

```
1  # VC0706 image capture to internal storage demo.
2  # You must wire up the VC0706 to the board's serial port, and enable writes
3  # to the internal filesystem by following this page to edit boot.py:
4  #   https://learn.adafruit.com/cpu-temperature-logging-with-circuit-python/writing-to-
   ↳ the-filesystem
5  import time
6
7  import board
8  import busio
9
10 import adafruit_vc0706
11
12 # Configuration:
13 IMAGE_FILE = '/image.jpg' # Full path to file name to save captured image.
14                          # Will overwrite!
15
16 # Setup SPI bus (hardware SPI).
17 spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
18
19 # Create a serial connection for the VC0706 connection, speed is auto-detected.
20 uart = busio.UART(board.TX, board.RX, timeout=250)
21 # Setup VC0706 camera
22 vc0706 = adafruit_vc0706.VC0706(uart)
23
24 # Print the version string from the camera.
25 print('VC0706 version:')
26 print(vc0706.version)
```

(continues on next page)

(continued from previous page)

```

27
28 # Set the image size.
29 vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480 # Or set VC0706_320x240 or
30                                                         # VC0706_160x120
31 # Note you can also read the property and compare against those values to
32 # see the current size:
33 size = vc0706.image_size
34 if size == adafruit_vc0706.IMAGE_SIZE_640x480:
35     print('Using 640x480 size image.')
36 elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
37     print('Using 320x240 size image.')
38 elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
39     print('Using 160x120 size image.')
40
41 # Take a picture.
42 print('Taking a picture in 3 seconds...')
43 time.sleep(3)
44 print('SNAP!')
45 if not vc0706.take_picture():
46     raise RuntimeError('Failed to take picture!')
47
48 # Print size of picture in bytes.
49 frame_length = vc0706.frame_length
50 print('Picture size (bytes): {}'.format(frame_length))
51
52 # Open a file for writing (overwriting it if necessary).
53 # This will write 50 bytes at a time using a small buffer.
54 # You MUST keep the buffer size under 100!
55 print('Writing image: {}'.format(IMAGE_FILE), end='')
56 with open(IMAGE_FILE, 'wb') as outfile:
57     wcount = 0
58     while frame_length > 0:
59         # Compute how much data is left to read as the lesser of remaining bytes
60         # or the copy buffer size (32 bytes at a time). Buffer size MUST be
61         # a multiple of 4 and under 100. Stick with 32!
62         to_read = min(frame_length, 32)
63         copy_buffer = bytearray(to_read)
64         # Read picture data into the copy buffer.
65         if vc0706.read_picture_into(copy_buffer) == 0:
66             raise RuntimeError('Failed to read picture frame data!')
67         # Write the data to SD card file and decrement remaining bytes.
68         outfile.write(copy_buffer)
69         frame_length -= 32
70         # Print a dot every 2k bytes to show progress.
71         wcount += 1
72         if wcount >= 64:
73             print('.', end='')
74             wcount = 0
75 print()
76 print('Finished!')

```

Listing 2: examples/snapshot.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

```

(continues on next page)

(continued from previous page)

```

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  import time
10
11 import board
12 import busio
13 import digitalio
14 import storage
15
16 import adafruit_sdcard
17 import adafruit_vc0706
18
19
20 # Configuration:
21 SD_CS_PIN = board.D10 # CS for SD card (SD_CS is for Feather Adalogger)
22 IMAGE_FILE = '/sd/image.jpg' # Full path to file name to save captured image.
23                               # Will overwrite!
24
25 # Setup SPI bus (hardware SPI).
26 spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
27
28 # Setup SD card and mount it in the filesystem.
29 sd_cs = digitalio.DigitalInOut(SD_CS_PIN)
30 sdcard = adafruit_sdcard.SDCard(spi, sd_cs)
31 vfs = storage.VfsFat(sdcard)
32 storage.mount(vfs, '/sd')
33
34 # Create a serial connection for the VC0706 connection, speed is auto-detected.
35 uart = busio.UART(board.TX, board.RX, timeout=250)
36 # Setup VC0706 camera
37 vc0706 = adafruit_vc0706.VC0706(uart)
38
39 # Print the version string from the camera.
40 print('VC0706 version:')
41 print(vc0706.version)
42
43 # Set the baud rate to 115200 for fastest transfer (its the max speed)
44 vc0706.baudrate = 115200
45
46 # Set the image size.
47 vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480 # Or set IMAGE_SIZE_320x240 or
48                                                         # IMAGE_SIZE_160x120
49 # Note you can also read the property and compare against those values to
50 # see the current size:
51 size = vc0706.image_size
52 if size == adafruit_vc0706.IMAGE_SIZE_640x480:
53     print('Using 640x480 size image.')
54 elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
55     print('Using 320x240 size image.')
56 elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
57     print('Using 160x120 size image.')
58
59 # Take a picture.
60 print('Taking a picture in 3 seconds...')

```

(continues on next page)

(continued from previous page)

```

61 time.sleep(3)
62 print('SNAP!')
63 if not vc0706.take_picture():
64     raise RuntimeError('Failed to take picture!')
65
66 # Print size of picture in bytes.
67 frame_length = vc0706.frame_length
68 print('Picture size (bytes): {}'.format(frame_length))
69
70 # Open a file for writing (overwriting it if necessary).
71 # This will write 50 bytes at a time using a small buffer.
72 # You MUST keep the buffer size under 100!
73 print('Writing image: {}'.format(IMAGE_FILE), end='')
74 with open(IMAGE_FILE, 'wb') as outfile:
75     wcount = 0
76     while frame_length > 0:
77         # Compute how much data is left to read as the lesser of remaining bytes
78         # or the copy buffer size (32 bytes at a time). Buffer size MUST be
79         # a multiple of 4 and under 100. Stick with 32!
80         to_read = min(frame_length, 32)
81         copy_buffer = bytearray(to_read)
82         # Read picture data into the copy buffer.
83         if vc0706.read_picture_into(copy_buffer) == 0:
84             raise RuntimeError('Failed to read picture frame data!')
85         # Write the data to SD card file and decrement remaining bytes.
86         outfile.write(copy_buffer)
87         frame_length -= 32
88         # Print a dot every 2k bytes to show progress.
89         wcount += 1
90         if wcount >= 64:
91             print('.', end='')
92             wcount = 0
93 print()
94 print('Finished!')

```

5.2 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

5.2.1 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 6

Indices and tables

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

a

adafruit_vc0706, [14](#)

A

adafruit_vc0706 (module), [14](#)

B

baudrate (adafruit_vc0706.VC0706 attribute), [15](#)

F

frame_length (adafruit_vc0706.VC0706 attribute), [15](#)

I

image_size (adafruit_vc0706.VC0706 attribute), [15](#)

R

read_picture_into() (adafruit_vc0706.VC0706 method),
[15](#)

T

take_picture() (adafruit_vc0706.VC0706 method), [15](#)

V

VC0706 (class in adafruit_vc0706), [14](#)

version (adafruit_vc0706.VC0706 attribute), [15](#)