Adafruit VC0706 Library Documentation

Release 1.0

Tony DiCola

Oct 20, 2019

Contents

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.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/vc0706_snapshot_internal.py

```
# VC0706 image capture to internal storage demo.
# You must wire up the VC0706 to the board's serial port, and enable writes
# to the internal filesystem by following this page to edit boot.py:
#   https://learn.adafruit.com/cpu-temperature-logging-with-circuit-python/writing-to-
↪the-filesystem
import time

import board
import busio

import adafruit_vc0706

# Configuration:
IMAGE_FILE = '/image.jpg' # Full path to file name to save captured image.
                          # Will overwrite!

# Setup SPI bus (hardware SPI).
spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)

# Create a serial connection for the VC0706 connection, speed is auto-detected.
uart = busio.UART(board.TX, board.RX, timeout=250)
# Setup VC0706 camera
vc0706 = adafruit_vc0706.VC0706(uart)

# Print the version string from the camera.
print('VC0706 version:')
print(vc0706.version)
```

(continues on next page)

(continued from previous page)

```

# Set the image size.
vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480 # Or set VC0706_320x240 or
                                                    # VC0706_160x120
# Note you can also read the property and compare against those values to
# see the current size:
size = vc0706.image_size
if size == adafruit_vc0706.IMAGE_SIZE_640x480:
    print('Using 640x480 size image.')
elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
    print('Using 320x240 size image.')
elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
    print('Using 160x120 size image.')

# Take a picture.
print('Taking a picture in 3 seconds...')
time.sleep(3)
print('SNAP!')
if not vc0706.take_picture():
    raise RuntimeError('Failed to take picture!')

# Print size of picture in bytes.
frame_length = vc0706.frame_length
print('Picture size (bytes): {}'.format(frame_length))

# Open a file for writing (overwriting it if necessary).
# This will write 50 bytes at a time using a small buffer.
# You MUST keep the buffer size under 100!
print('Writing image: {}'.format(IMAGE_FILE), end='')
with open(IMAGE_FILE, 'wb') as outfile:
    wcount = 0
    while frame_length > 0:
        # Compute how much data is left to read as the lesser of remaining bytes
        # or the copy buffer size (32 bytes at a time).  Buffer size MUST be
        # a multiple of 4 and under 100.  Stick with 32!
        to_read = min(frame_length, 32)
        copy_buffer = bytearray(to_read)
        # Read picture data into the copy buffer.
        if vc0706.read_picture_into(copy_buffer) == 0:
            raise RuntimeError('Failed to read picture frame data!')
        # Write the data to SD card file and decrement remaining bytes.
        outfile.write(copy_buffer)
        frame_length -= 32
        # Print a dot every 2k bytes to show progress.
        wcount += 1
        if wcount >= 64:
            print('.', end='')
            wcount = 0
print()
print('Finished!')
```

Listing 2: examples/vc0706_snapshot_simpletest.py

```
# VC0706 image capture to SD card demo.
# You must wire up the VC0706 to the board's serial port, and a SD card holder
# to the board's SPI bus.  Use the Feather M0 Adalogger as it includes a SD
```

(continues on next page)

(continued from previous page)

```
# card holder pre-wired to the board--this sketch is setup to use the Adalogger!
# In addition you MUST also install the following dependent SD card library:
#   https://github.com/adafruit/Adafruit_CircuitPython_SD
# See the guide here for more details on using SD cards with CircuitPython:
#   https://learn.adafruit.com/micropython-hardware-sd-cards
import time

import board
import busio
import digitalio
import storage

import adafruit_sdcard
import adafruit_vc0706


# Configuration:
SD_CS_PIN = board.D10  # CS for SD card (SD_CS is for Feather Adalogger)
IMAGE_FILE = '/sd/image.jpg'  # Full path to file name to save captured image.
                              # Will overwrite!

# Setup SPI bus (hardware SPI).
spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)

# Setup SD card and mount it in the filesystem.
sd_cs = digitalio.DigitalInOut(SD_CS_PIN)
sdcard = adafruit_sdcard.SDCard(spi, sd_cs)
vfs = storage.VfsFat(sdcard)
storage.mount(vfs, '/sd')

# Create a serial connection for the VC0706 connection, speed is auto-detected.
uart = busio.UART(board.TX, board.RX, timeout=250)
# Setup VC0706 camera
vc0706 = adafruit_vc0706.VC0706(uart)

# Print the version string from the camera.
print('VC0706 version:')
print(vc0706.version)

# Set the baud rate to 115200 for fastest transfer (its the max speed)
vc0706.baudrate = 115200

# Set the image size.
vc0706.image_size = adafruit_vc0706.IMAGE_SIZE_640x480 # Or set IMAGE_SIZE_320x240 or
                                                       # IMAGE_SIZE_160x120
# Note you can also read the property and compare against those values to
# see the current size:
size = vc0706.image_size
if size == adafruit_vc0706.IMAGE_SIZE_640x480:
    print('Using 640x480 size image.')
elif size == adafruit_vc0706.IMAGE_SIZE_320x240:
    print('Using 320x240 size image.')
elif size == adafruit_vc0706.IMAGE_SIZE_160x120:
    print('Using 160x120 size image.')

# Take a picture.
print('Taking a picture in 3 seconds...')
```

(continues on next page)

(continued from previous page)

```
time.sleep(3)
print('SNAP!')
if not vc0706.take_picture():
    raise RuntimeError('Failed to take picture!')

# Print size of picture in bytes.
frame_length = vc0706.frame_length
print('Picture size (bytes): {}'.format(frame_length))

# Open a file for writing (overwriting it if necessary).
# This will write 50 bytes at a time using a small buffer.
# You MUST keep the buffer size under 100!
print('Writing image: {}'.format(IMAGE_FILE), end='')
with open(IMAGE_FILE, 'wb') as outfile:
    wcount = 0
    while frame_length > 0:
        # Compute how much data is left to read as the lesser of remaining bytes
        # or the copy buffer size (32 bytes at a time).  Buffer size MUST be
        # a multiple of 4 and under 100.  Stick with 32!
        to_read = min(frame_length, 32)
        copy_buffer = bytearray(to_read)
        # Read picture data into the copy buffer.
        if vc0706.read_picture_into(copy_buffer) == 0:
            raise RuntimeError('Failed to read picture frame data!')
        # Write the data to SD card file and decrement remaining bytes.
        outfile.write(copy_buffer)
        frame_length -= 32
        # Print a dot every 2k bytes to show progress.
        wcount += 1
        if wcount >= 64:
            print('.', end='')
            wcount = 0
print()
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

Python Module Index

Index