

Handling a Drone Camera with Arduino

Fernando G. Tinetti*

Technical Report TR-RT-02-2019
III-LIDI, Fac. de Informática, UNLP
CIC, Prov. de Buenos Aires
Argentina
contact e-mail: fernando@info.unlp.edu.ar
June 2019

Abstract. This technical report documents how to configure the power and generate the signal/s required to handle a spare Syma X5C drone camera. This camera model is very simple (as most camera drones, actually), with a few reports of being used as a standalone picture and recording camera. We show how to power the camera as well as generate the signals from an Arduino development board for taking pictures as well as for video recording.

1.- Introduction

Several low-cost and toy-like drones such as the Syma X5C drone [2] include a small and simple camera. All of them usually also include very simple control signals for taking pictures and video recording. In particular, the Syma X5C drone camera is one of the most basic cameras, which only records images/videos in a micro-SD memory, and does not include any transmission facility. Even more, the interconnection between the camera and the drone controller is reduced to only three wires, as shown in Fig. 1: Vcc (Red wire), Gnd (Black wire), and Signal (Yellow wire).



Figure 1: Syma X5C Drone Camera.

There are a few web sites/videos documenting the usage of this or a similar camera as a stand-alone camera, and we will use it in a similar way, i.e. without the drone itself. However, we will include a simple Arduino control sketch as a starting point for controlling the camera.

2.- Power and Signal Wires

As explained before, the camera has three wires: two of them for power and one for signal/camera control behavior. The power wires, Vcc and Gnd should be used for turning on the camera, there is no power button. The first unknown detail about power source is related to the voltage level. A priori, it seems to be the most important one, because there is a risk of damaging the camera if provided with a too high Vcc. We will use the drone source power as the strongest reference of information, and the drone we have access to is powered from a pair of 3.7 Li-Po batteries connected in parallel. We will assume the Vcc needed by the drone camera is not far from the 1S Li-Po battery power, and maybe the camera is directly powered from the 1S battery pack. Given that 3.7v is not easily obtained from any Arduino development board, we will use the “standard” 3.3v output from almost every (if not all) Arduino development board. Actually, we do not suggest this power source to the camera as the “best one” not even suggested in general, this is what we have used and we document in this technical report. We will assume that if the 3.3v and/or the current provided by the Arduino is not enough for the camera, it will not work (and it will not be damaged). Thus, Fig. 2 show the power source to the camera from the Arduino pins.

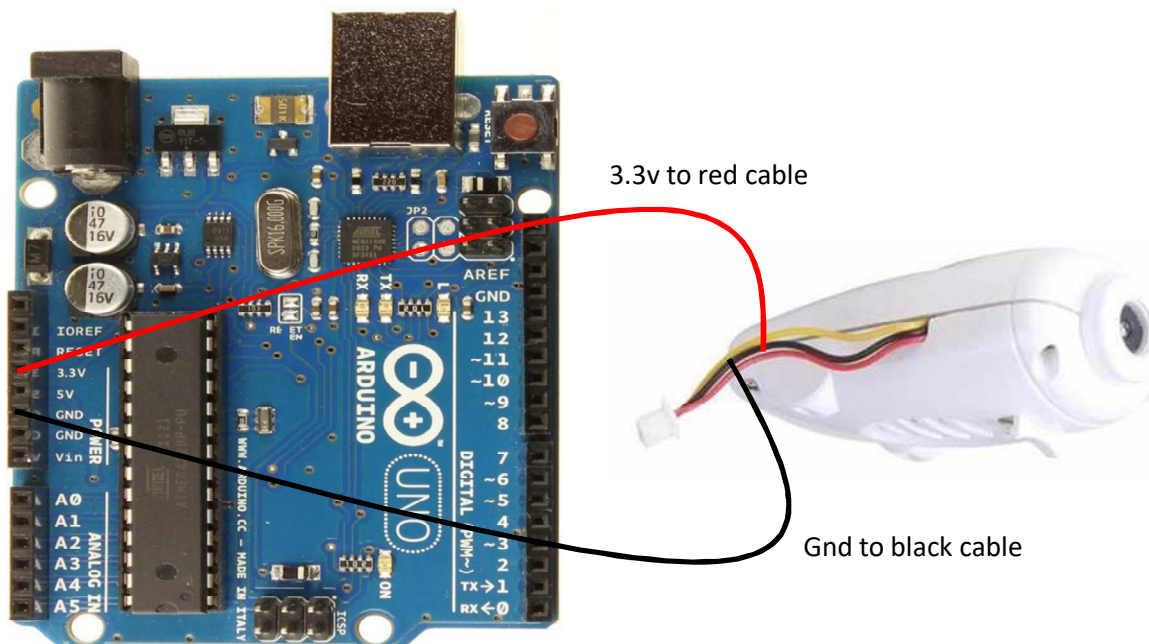


Figure 2: Camera Power from Arduino Pins.

At this point, just powering up the Arduino development board the camera also turns on its led/s, the 3.3v and Gnd Arduino pins seem to be enough to power up the camera. However, we have to generate the signal/s in the yellow wire in order to verify the camera is actually working.

The camera works only if the micro-SD memory card is mounted, otherwise it only turns on its leds. If the power is provided, and the micro-SD memory card is mounted, then the camera is able to take pictures and recording video. The camera identifies the signal/s in the Signal wire (the yellow one) as the control signal/s for its operation:

- Nothing is done while the Signal wire is not connected any voltage level.
- A picture is taken if there is a short period of time in which the yellow wire is connected to Gnd.
- Video recording is started if there is a relatively long period of time in which the yellow wire is connected to Gnd.

- If the camera is recording video, it ends recording if the yellow wire is connected to Gnd one again.

We made a few and preliminary experiments just plugin in the yellow wire to an Arduino Gnd pin and we have found that

- If the yellow wire is plugged in to Gnd by a fraction of a second (a quick manual plug-unplug action) the camera takes a picture.
- If the yellow wire is plugged in to Gnd by 1 or 2 seconds, the camera starts recording video, and it ends recording video as soon as the yellow wire is connected to Gnd once again.

We were able to see the pictures and the video recorded in the microSD card plugged in the camera. The previous experiments provided a manual and rough way of verifying that

- The camera is able to work with the 3.3v from the Arduino development board
- Connecting the yellow wire to Gnd in a “controlled” way it is possible to define the camera operation: a) take picture, b) start video recording, and b) end video recording.

Thus, controlling the “information” in the Signal wire implies controlling the camera, i.e. at least generating the Gnd signal in to the yellow wire from the Arduino development board.

3.- Arduino Control

The basic sequence we have followed for the Signal wire is

- Set the pin connected to the Signal wire as INPUT, and Arduino pins configured this way are said to be in a high-impedance state [1].
- For generating a Gnd in the Signal wire pin, the pin is connected as OUTPUT and a LOW signal is written by a specified amount of time, after which the pin is set back as INPUT.
- The amount of time we have verified that works as a “short period of time” is 200ms and the one that works as a “long period of time” is 2s.

Fig. 3 shows the three basic Arduino functions for handling the camera, where SIGNALPIN is the pin connected to the camera yellow wire, and SHORTDELAY and LONGDELAY are the corresponding delay values, i.e. 200 and 2000 respectively. It is worth noting that the identification of either a short or long period of time in Gnd state of the Signal value is relative to the high-impedance state, as if the wire is not connected to anything. This is why the pin is initialized as INPUT at the beginning of the execution as well as the pin is set as INPUT once again after it has been set as OUTPUT with a LOW (Gnd) signal value. Taking advantage of the functions shown in Fig. 3, we have also defined more meaningful camera control functions names as well as minimum inter-pictures and video recording time, shown in Fig. 4, i.e.:

- camera_begin(), for initially setting the pin as INPUT (i.e. in high impedance state).
- picture(), for taking a picture, which is in fact a call to shortGND() plus a delay of 1s
- startVideo(), for starting video record, which is in fact a call to longGND() plus a delau of 1s (thus, every video will be at least 1s long).
- stopVideo(), for ending video record, which is in fact a call to longGND() plus a delau of 1s.

```

/*****/
void camera_begin(){
  pinMode(SIGNALPIN, INPUT);
}
/*****/
void shortGND(){
  pinMode(SIGNALPIN, OUTPUT);
  digitalWrite(SIGNALPIN, LOW);
  delay(SHORTDELAY);
  // Pin set as input ==> hig impedance state... end of GND signal
  pinMode(SIGNALPIN, INPUT);
}
/*****/
void longGND(){
  pinMode(SIGNALPIN, OUTPUT);
  digitalWrite(SIGNALPIN, LOW);
  delay(LONGDELAY);
  // Pin set as input ==> hig impedance state... end of GND signal
  pinMode(SIGNALPIN, INPUT);
}
/*****/

```

Figure 3: Arduino Basic Camera Control Functions.

```

/*****/
void picture()
{
  shortGND();
  delay(1000); // ==> 1 sec. after picture
}
/*****/
void startVideo()
{
  longGND();
  delay(1000); // ==> 1 sec. minimum video recording
}

/*****/
void stopVideo()
{
  shortGND();
  delay(1000); // ==> 1 sec. after video recording
}
/*****/

```

Figure 4: Arduino Camera Control Functions.

4.- WiFi Control

Once we are able to handle the signals for controlling the camera via the Arduino environment it is easy to have the same control via WiFi by using the ESP8266 WiFi module [3]. As explained in [3], the WiFi module is basically programmed as an Arduino with the added facility of having a WiFi module directly available. Fig. 5 shows the web interface added with a few source code lines added to those already shown in a Wemos development board. The general structure of the code in the Wemos, which is also programmed in the Arduino IDE is:

- Setup of the ESP8266 as an access point.
- Provide the simple index.html shown in Fig. 4 (actually, a single html form) with the corresponding HTTP server running as an Arduino process.
- Process the button selected in the form and call the corresponding camera function from those shown in Fig. 4 above.

Besides, the project structure with

- ESP8266 WiFi configured as Access Point
- Set a HTTP server for providing a web control page as a HTML form or even with AJAX
- Process the client request for controlling a system (the camera in this report, as example)

is useful for a wide number of control, monitoring, and control and monitoring applications. The Arduino environment along with the ESP8266 module would be able to handle most (if not all) interactive applications as well as many real-time monitoring and control applications. One of the limiting factors would be the number of pins available, which in the case of this report is only one: the pin connected to the camera Signal wire (the yellow one).

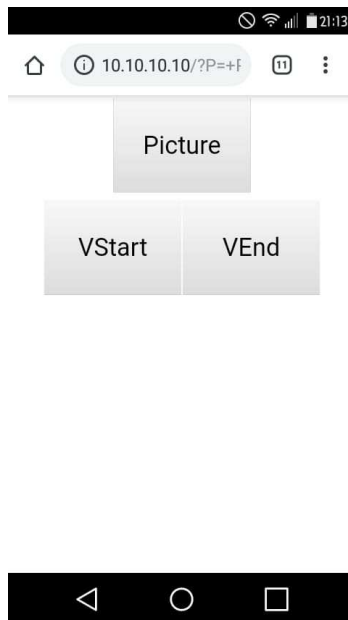


Figure 5: Controlling the Camera via ESP8266 WiFi (Wemos Development Board).

References

- [1] Arduino - DigitalPins (available June 2019)
<https://www.arduino.cc/en/Tutorial/DigitalPins>

[2] Syma, SYMA X5C EXPLORERS - Drone - SYMA Official Site (available June 2019),
<http://www.symatoys.com/goodshow/x5c-syma-x5c-explorers.html>

[3] Fernando G. Tinetti, "Programming (Flashing) and Using the ESP8266-(01/07)", Technical Report TR-RT-01-2019, III-LIDI/CIC Prov. de Bs. As., Fac. de Informática, UNLP, June 2019 (available June 2019)
<http://fernando.thats.im/links/embed-rt/2019-ESPFlash.pdf>