

## Building a photo booth by using ARTIK 10

In the most recent ARTIK binary release, we began to roll out multimedia features. Here I want to show you how you can build a voice controlled photo booth by using ARTIK 10. We will use voice commands to trigger the camera and display the photo taken on a HDMI display that connects to an ARTIK 10.

Hardware required:

- An ARTIK 10 board running 1020GC0F-3AF-01Q2 image
- A USB camera
- A HDMI display

Connect the USB camera and the display to your ARTIK 10 board

Software:

We will use Python for our photo booth development. Python v2.7.10 is pre-loaded with the latest ARTIK 10 binary.

### Before we start

Please make sure the binary version on your ARTIK 10 supports multimedia features. You can run 'cat /etc/artik\_release' from your ARTIK command line.

```
# cat /etc/artik_release
RELEASE_VERSION=1020GC0F-3AF-01Q2
RELEASE_DATE=20160418.203719
RELEASE_UBOOT=U-Boot 2012.07
RELEASE_KERNEL=3.10.93
```

The 2<sup>nd</sup> line of the output shows the release date of your binary. If your binary version is earlier than Apr 18th, please update it to the latest by following our guideline at <https://developer.artik.io/documentation/developer-guide/updating-artik-image.html>. You can find the latest ARTIK 10 image at <https://developer.artik.io/downloads> (You need to sign in to download).

We will use the sample codes in our programming guides as the basic building blocks for our photo booth. Please refer to Audio Programming Guide on how to use Python to control ARTIK audio:

<https://developer.artik.io/documentation/developer-guide/multimedia/audio-guide.html>

ARTIK 10 USB camera Programming guide explains how you can use OpenCV Python bindings to take pictures.

<https://developer.artik.io/documentation/developer-guide/multimedia/usb-camera.html>

ARTIK Display Guide lists in detail the steps of configuring a HDMI display and showing an image on it.

<https://developer.artik.io/documentation/developer-guide/multimedia/display.html>

### Let's start coding

I am using Google Voice Recognition APIs to process the voice commands. Since it is a cloud-based solution, please make sure your ARTIK board is online. We also need to install some packages on our ARTIK board.

```
#dnf install python-devel portaudio portaudio-devel flac flac-devel
#pip install PyAudio SpeechRecognition
```

You may be prompted to upgrade pip. Just run the command below to update:

```
#pip install --upgrade pip
```

Here is our speech recognition code *photobooth.py*. The code is adapted from <https://pypi.python.org/pypi/SpeechRecognition/>. As you can see, you can easily change the code to use other voice recognition engines.

```
import speech_recognition as sr

#obtain audio from the microphone
r = sr.Recognizer()
while True:
    with sr.Microphone() as source:
        print("Say something to ARTIK!")
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)

    # recognize speech using Google Speech Recognition
    try:
        print("You said " + r.recognize_google(audio))
    except sr.UnknownValueError:
        print("Google Speech Recognition could not understand audio")
    except sr.RequestError as e:
        print("Could not request results from Google Speech Recognition
service: {0}".format(e))
```

Now, if you run the code from command line, you will see output like below (Warning messages about not connecting to the server can be ignored; they refer to real-time sound servers like [jackd](#) that are not needed.).

```
#python photobooth.py
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
Say something to ARTIK!
Speech Recognition thinks you said take a picture
```

We can then create a `take_a_photo_and_display()` function, which takes a photo and display it on the attached screen.

```
import cv2
import subprocess

def take_a_photo_and_display():
    cap = cv2.VideoCapture(0)
    ret, frame = cap.read()
```

```

cv2.imwrite("photo.jpg", frame)
cap.release()
subprocess.call("fbi -T 2 photo.jpg", shell=True)

```

We will use “take a picture” as trigger words to trigger the camera action. If we combine the voice recognition code and photo taking and displaying code, we will have a working photo booth:

```

import speech_recognition as sr
import subprocess
import cv2

# take a photo
def take_a_picture_and_display():
    cap = cv2.VideoCapture(0)
    ret, frame = cap.read()
    cv2.imwrite("photo.jpg", frame)
    cap.release()
    subprocess.call("fbi -T 2 photo.jpg", shell=True)

# obtain audio from the microphone
r = sr.Recognizer()
while True:
    with sr.Microphone() as source:
        print("Say something to ARTIK!")
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)

# recognize speech using Google Speech Recognition
try:
    string = r.recognize_google(audio)
    print("You said: " + string)
    if "take a picture" in string:
        take_a_picture_and_display()
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
    print("Could not request results from Google Speech Recognition service: {0}".format(e))

```

Now, if you say “take a picture” when you get promoted, ARTIK will take a picture for you and display it on the HDMI display.