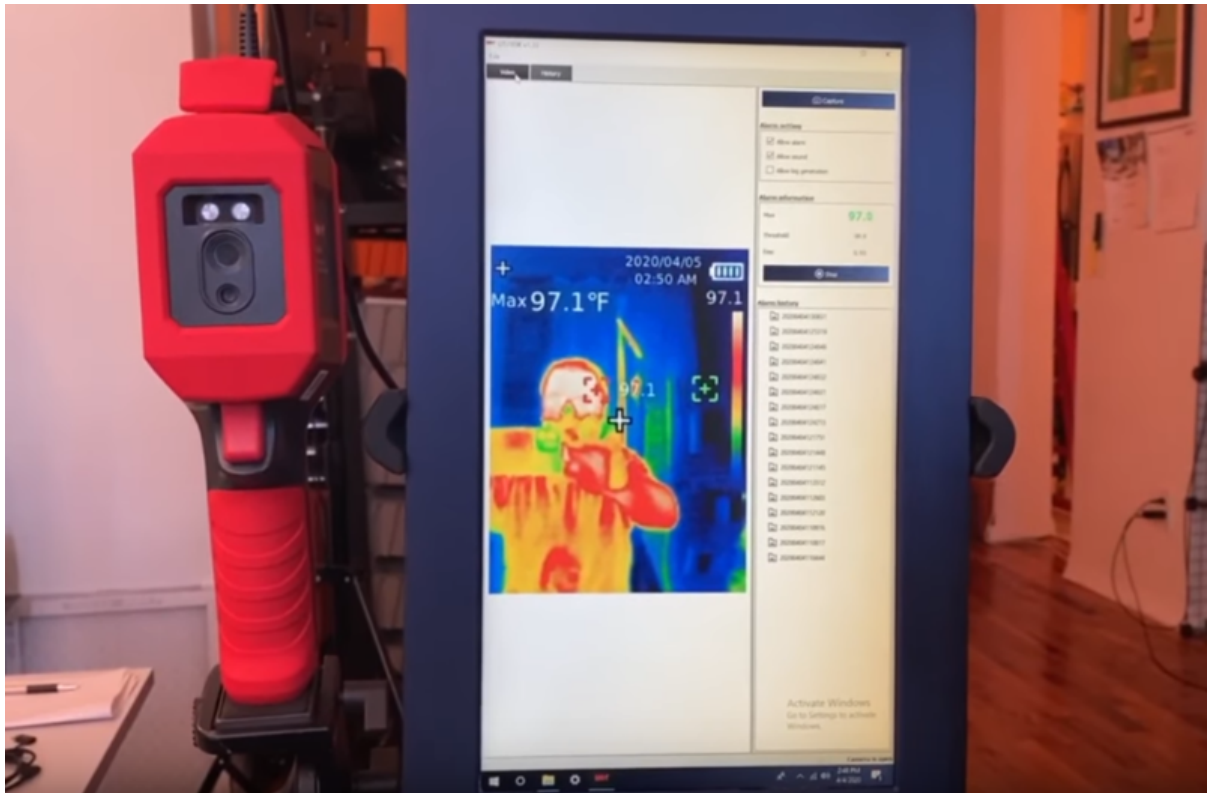




UTi165 Thermal Fever Scanner Camera

Created by lady ada



<https://learn.adafruit.com/uti165-thermal-fever-scanner-camera>

Last updated on 2024-06-03 03:06:34 PM EDT

Table of Contents

Overview	3
Default Windows Software	4
<ul style="list-style-type: none">• Change Language Option• Running the App	
Python Software	8
Alternative Viewers	11

Overview



This video camera takes photos of temperatures! This camera is specifically tuned to work in the 30°C~45°C / 86°F~113°F range with 0.5°C / 1°F accuracy, so it's excellent for human temperature & fever detection. In fact, this thermal camera is often used by companies/airports/hotels/malls to do a first-pass fever check: If any person has a temperature of over 99°F an alarm goes off so you can do a secondary check with an accurate handheld temperature meter.

You may have seen thermal 'FLIR' cameras used to find air leaks in homes, but those cameras have a very wide temperature range, so they're not as accurate in the narrow range used for fever-scanning. This camera is designed specifically for that purpose!

There's a color 2.8" 320x240 TFT screen which gives real-time temperature readings from an interpolated 160x120 thermal resolution, with hot-spot temperatures labeled. You can also set temperature ranges for alerts, F or C units, color palettes, etc. There's a built-in li-poly battery so it's fully portable. You can recharge it over USB.

We like these fever scanners because:

- They are surprisingly accurate
- Work great at up to ~10 meter distance
- No Internet connection needed - completely standalone functionality
- [Can be connected to any Windows computer for a larger display with free available software \(video here\) \(https://adafru.it/KDw\)](https://adafru.it/KDw)

- [Can be used to capture thermal data with open-source software \(in development\) \(https://adafru.it/KDx\)](https://adafru.it/KDx)
- Can run on rechargeable battery for many hours or plugged in
- Works as good as "\$5000 machine learning" setups at a fraction of the cost

The UTi165K version covered in this guide has real-time data transmission to a computer for analysis over USB. It also features three viewing modes: Thermal (infrared image), Digital (regular camera), and Fusion (a blend of the two). We also carry the [UTi165H](http://adafru.it/4578) (<http://adafru.it/4578>), which has only a thermal viewing mode and no USB output.



Thermal Camera Imager for Fever Screening with USB Video Output

This video camera takes photos of temperatures! This camera is specifically tuned to work in the 30°C~45°C / 86°F~113°F range with 0.5°C / 1°F accuracy, so it's excellent for...

<https://www.adafruit.com/product/4579>

You can also capture images and measurements and transfer them to your computer via USB type C or SD Card.

Default Windows Software



Uni-T has released software for the UTi165K thermal scanner with video out. It's very simple but works for most use cases.

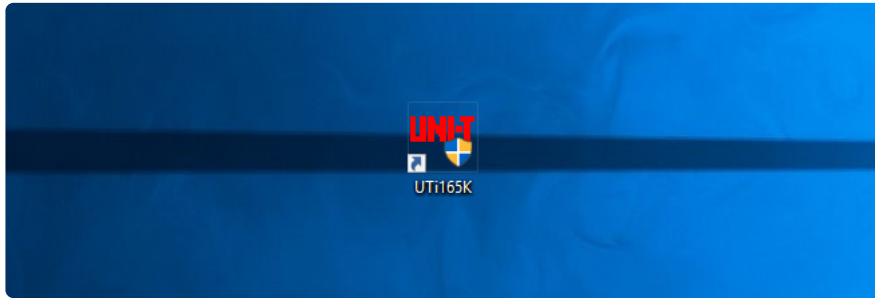
This software is Windows only!

Download it by clicking this button:

4579_UTi165K_Software_Win.zip

<https://adafru.it/KDz>

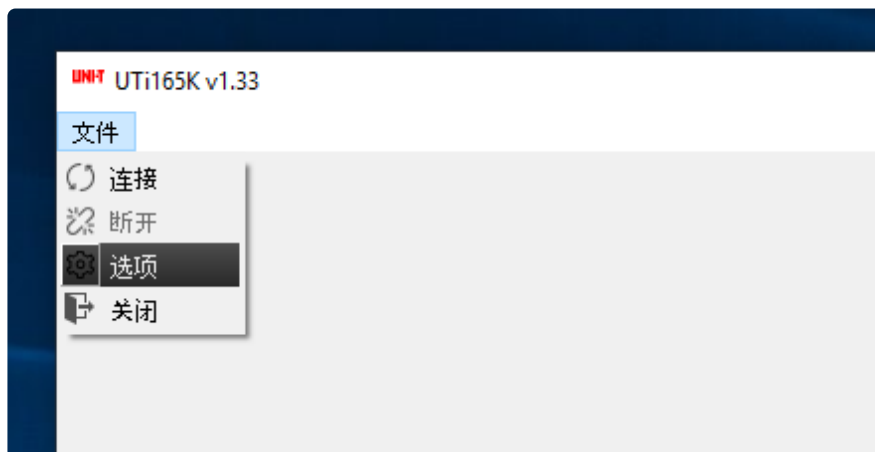
Un-zip and run the installer to get the desktop icon for the app



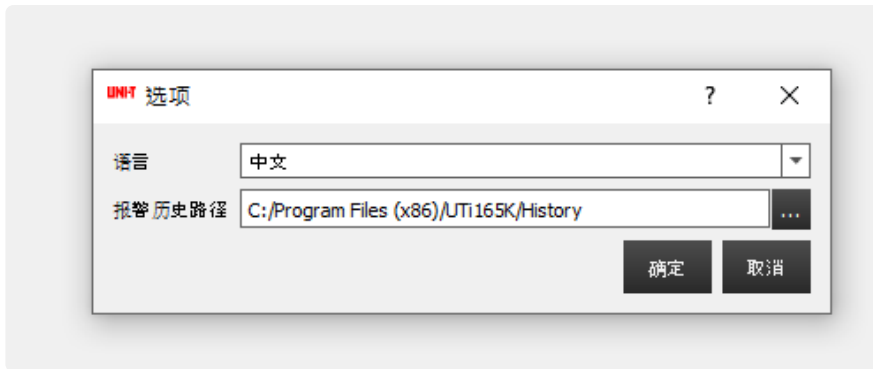
Change Language Option

By default the language the fever scanner comes in Chinese. If you read Chinese, great! If not...here's how to change it to English

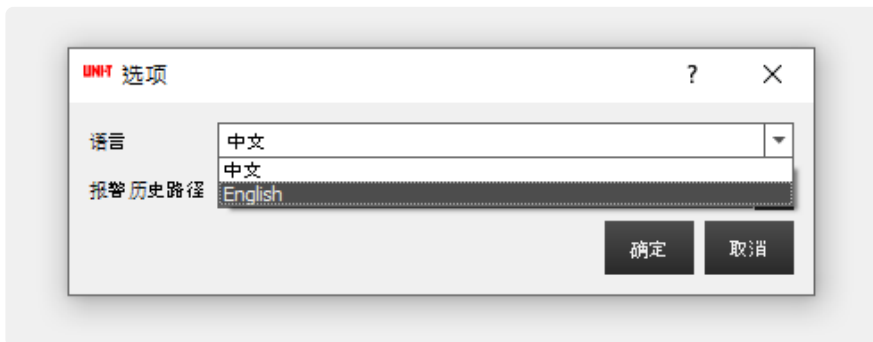
Unplug the scanner from USB before starting the app. Then open the app and look for the menu in the top left, select the **GEAR ICON** (third icon)



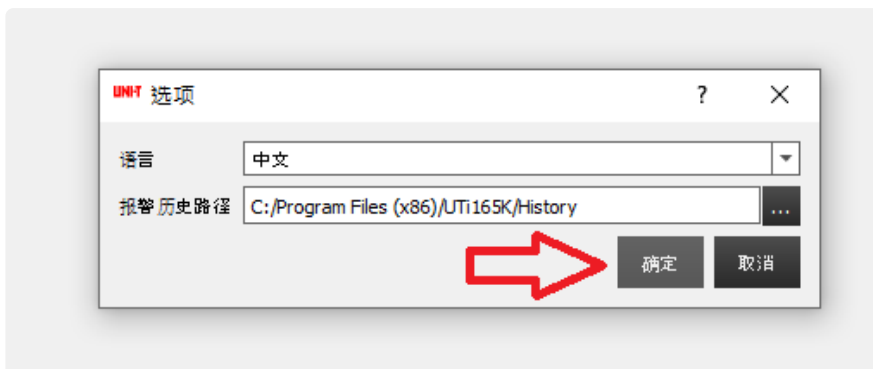
This dialog will open up



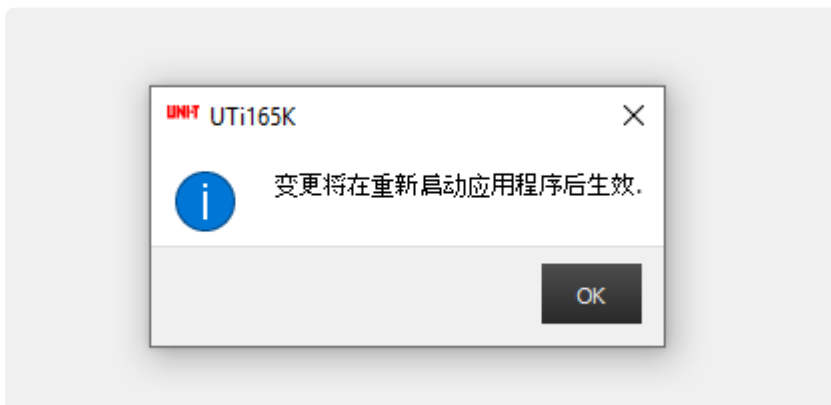
From the drop down, select **English**



Press the LEFT button



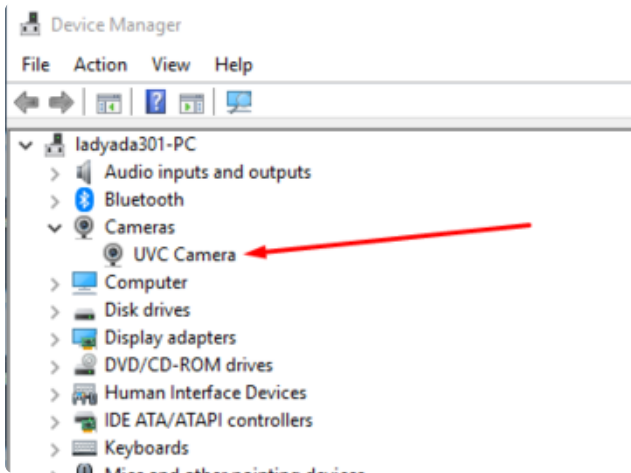
You'll get this notice (it says changes will take affect on next launch).



Quit the app and restart it. This time you can plug in the camera before you start.

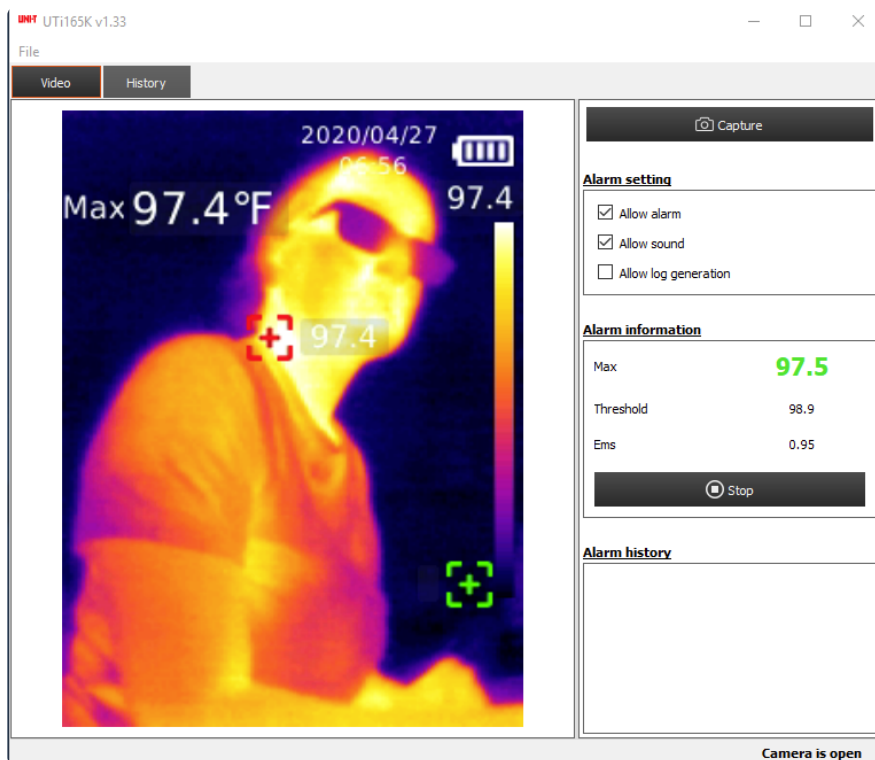
Running the App

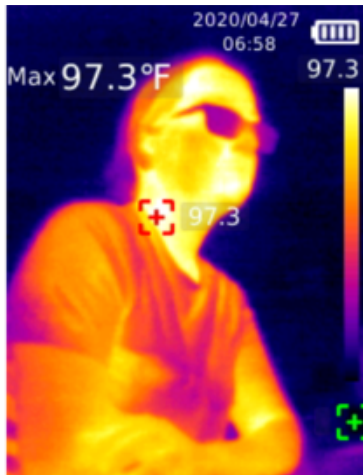
Window/the app sometimes get a little confused if you've turned on/off the camera or disconnected it. If you can't get the app to connect: quit the app, disconnect the USB C cable, power down the camera, wait 5 seconds, turn it back on, re-plug in the USB cable and start the app. Once its running, you won't have any issue - it just seems to sometimes occur on initial connection.



Also make sure the camera shows up in the **Device Manager** as a generic **UVC Camera**

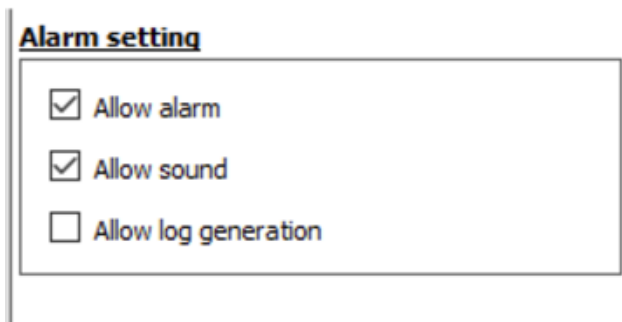
Once connected, you'll get a video with a mirror of the display.





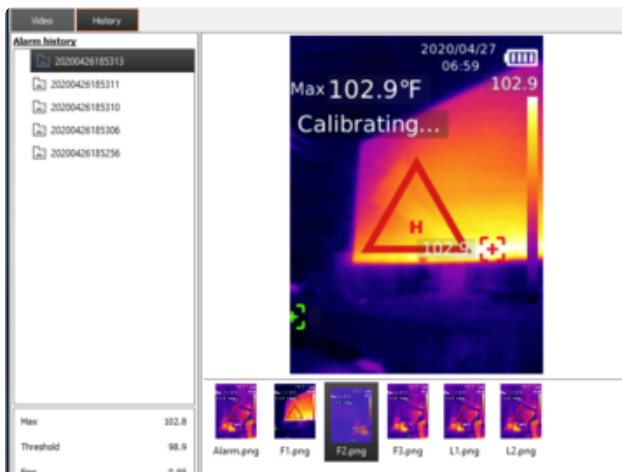
If you press the **Capture button** in the top left you'll get a PNG download

There's a few alarm options.



Allow sound means that the computer will beep (not the device, the computer speakers) when temperature goes above the upper set limit.

Log generation will take photos every time a high temperature event occurs



Click on the **History** tab to see all the images captured when temperature went above the upper limit

Python Software

The Windows software is distributed as an exe, and is not available for Mac or Linux. In order to use with other platforms, you can use Python and [opencv \(https://adafru.it/KEY\)](https://adafru.it/KEY), which [has interface hooks to read video camera frames \(https://adafru.it/KEZ\)](https://adafru.it/KEZ).

We tried this with Mac and Windows, on Windows we had to upgrade to the latest 4.3.0.36 opencv python package

Try out this example code, which will fetch frames and display them in opencv. Nothing else fancy is done, but you can build on this example!

```
import numpy as np
import time
import cv2
import struct

camera_num = 0

for camera_num in range(6):
    cam = cv2.VideoCapture(camera_num)
    if not cam.isOpened():
        print("Was not able to open camera", camera_num)
        cam.release()
        continue
    if not cam.set(3, 240):
        print("Was not able to set camera", camera_num, "width to 240 pixels")
        cam.release()
        continue
    if not cam.set(4, 321):
        print("Was not able to set camera", camera_num, "height to 321 pixels")
        cam.release()
        continue
    if cam.get(3) != 240:
        print("Was not able to set camera", camera_num, "width to 240 pixels")
        cam.release()
        continue
    break

print("Camera %d open at size: (%d x %d) %d FPS" % (camera_num, cam.get(3),
cam.get(4), cam.get(5)))

cv2.namedWindow('Thermal Camera - Press Q to quit', cv2.WINDOW_NORMAL)
cv2.resizeWindow('Thermal Camera - Press Q to quit', 480, 642)

while(True):
    # Capture frame-by-frame
    ret, frame = cam.read()

    if not ret:
        print("Failed to fetch frame")
        time.sleep(0.1)
        continue
    #print("Frame OK!")
    colorframe = cv2.cvtColor(frame, cv2.COLOR_BGR2BGR)
    # Display the resulting frame
    cv2.imshow('Thermal Camera - Press Q to quit', colorframe)

    cam.set(cv2.CAP_PROP_CONVERT_RGB, 0)
    ret, frame = cam.read()
    if not ret:
        print("Failed to fetch frame")
        time.sleep(0.1)
        continue
    print("Temp calculation (experimental): ", end="" )
    print(struct.unpack("h", frame[320][0][0:2])[0]/10)
    cam.set(cv2.CAP_PROP_CONVERT_RGB, 1)

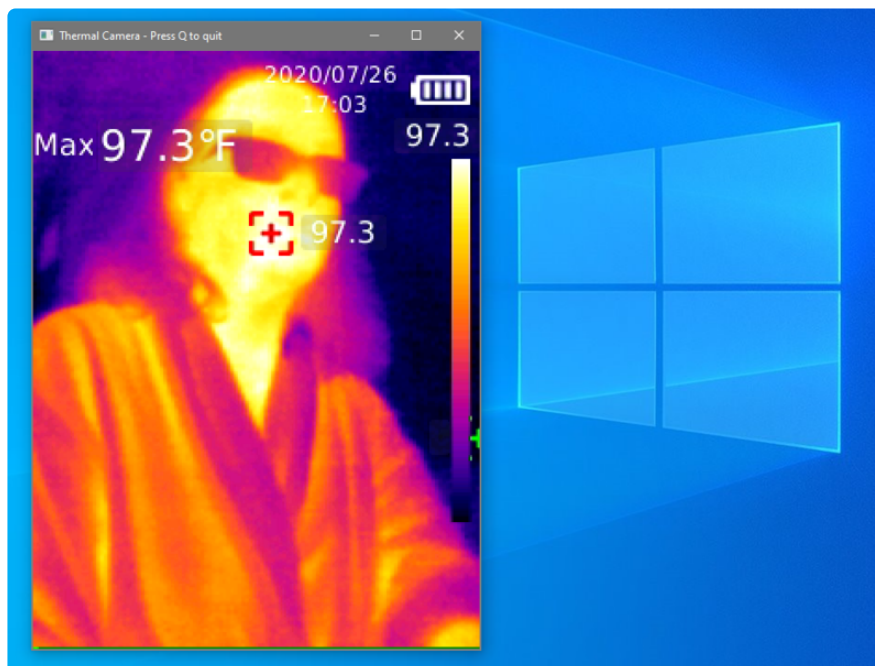
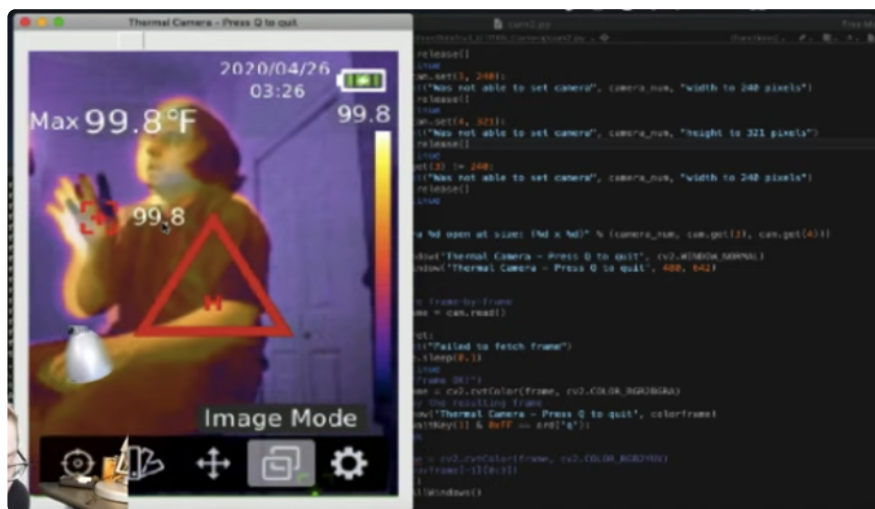
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

```
#yuvframe = cv2.cvtColor(frame, cv2.COLOR_RGB2YUV)
#print(yuvframe[-1][0:3])
cam.release()
cv2.destroyAllWindows()
```

Run it at the command line with the camera plugged in!

```
% python opencv_util165k.py
```

You may need to [install opencv python package \(https://adafru.it/KE-\)](https://adafru.it/KE-) first with `pip install opencv-python`



The output will be exactly what is on the scanner's TFT display.

Alternative Viewers

If you don't want to run the default Windows software, you can use a video cam viewer.

Note that many basic webcam viewers won't work because of the unusual 240x321 resolution!

Here's a few apps we tried that did work, but we don't have any guarantee or recommendation that these are any good, secure, or reliable:

- [SFT-15010 Viewer Software \(https://adafru.it/KDA\)](https://adafru.it/KDA)

If you have other video software suggestions, please open an issue at https://github.com/adafruit/Adafruit_UT1165_Camera (<https://adafru.it/KDx>) to let us know!