

# FADU Enlarger Timer

[Version 1.2.0 01-Feb-2019]

## Introduction

This is the combined User and Technical Manual for the FADU Enlarger Timer app for Android devices.

The timer can be used manually as a way of generating f-stop adjusted times for use with a separate enlarger timer (or simple on/off switch!) or can optionally be connected to a WiFi enabled Smart Socket to switch the enlarger lamp. There are two options to drive external hardware:

- 1) Sonoff S20 smart-socket which can be plugged into the mains directly with the enlarger plugged into the Sonoff.
- 2) A widely available relay module board which can be used to switch the low-voltage side of an enlarger's transformer/voltage regulator.

- 1) In the case of the Sonoff S20, unfortunately, I have been unable to find any commercial WiFi smart-socket that can connect *directly* to an Android device – all I have found must connect via a Cloud-based service and this is not suitable for our purpose due to the inherent unpredictable time delays of this connection method.

This means that we will need to reprogram the Sonoff device. This smart-socket is available widely in local mains voltage options and has a large online community of makers and hobbyists who customise the device with new firmware. It is not difficult for the technically-minded to program, but does require soldering a connector to existing pads on the circuit board to connect your computer to the socket via an FTDI adaptor (widely available on ebay or Amazon very cheaply) in order to download the code.

Details of how to reprogram the Sonoff S20 with links to online references and examples I used are in Appendix 3 of this manual.

The code for this app and for reprogramming the Sonoff S20 is all freely available and open to anyone who wishes to use it.

The Sonoff S20 is being replaced by the Sonoff S26 by the manufacturer which requires a slightly different method to reprogram but uses the same code.

- 2) An alternative option is to use a WiFi relay board available from LC Technologies ([http://www.chinalctech.com/index.php?m=mod\\_product&a=view&p\\_id=1261](http://www.chinalctech.com/index.php?m=mod_product&a=view&p_id=1261)) which is widely available. This does not require reprogramming so will work directly with the darkroom timer app without modification, but it is an unboxed circuit board module so will need boxing and connectors added as required. Also, I **STRONGLY** advise against using this board with mains voltages. The physical separation between the relay terminal traces and the rest of the circuit on this board is far too short and consider it highly unsuitable for use with high voltages. This board should only be used on the low-voltage side of a transformer as is usually supplied with

colour enlargers – typically 12 or 24 volts dc. Additionally, lamps greater than 150 watts at 24 volts (or 300w at 12V) should not be used. See Appendix 1 for details.

I bought mine from Banggood ([https://www.banggood.com/5V-ESP8266-Dual-WiFi-Relay-Module-Internet-Of-Things-Smart-Home-Mobile-APP-Remote-Switch-p-1270421.html?rmmds=detail-left-hotproducts\\_\\_2&HotRecToken=CgEwEAlaAklWlgJQRCgB&cur\\_warehouse=CN](https://www.banggood.com/5V-ESP8266-Dual-WiFi-Relay-Module-Internet-Of-Things-Smart-Home-Mobile-APP-Remote-Switch-p-1270421.html?rmmds=detail-left-hotproducts__2&HotRecToken=CgEwEAlaAklWlgJQRCgB&cur_warehouse=CN) ). I did buy one from a random ebay seller which also worked OK but there are no guarantees that they will have the same firmware installed (to be fair, there is no guarantee one bought 6 months later from Banggood will have the same firmware!).

## Release Notes

### 1.2:

1. **Stay Awake:** Screen no longer goes to sleep (assuming your phone allows sleeping to be overridden).
2. **Manual Link:** added a button in the Settings menu to open this user manual on the FADU forum site using the phone's default web browser.
3. **WiFi Relay Driver:** Can now optionally drive an LC Technologies WiFi relay module for low-voltage use. No reprogramming required for this module but it should only be used on the **low voltage side** of a colour enlarger's transformer/voltage stabiliser. It will also need a box and a 5V dc power supply.
4. **Count-in Option:** an optional 3-second delay to the count start added to allow time to prepare before the count starts.

### 1.1:

1. **1/10 Second Timing:** Added ability to count in 0.1 second increments.
  - 1.1. **Up/Down buttons removed.** Setting the count is now performed by tapping the top (increment) or bottom (decrement) half of the required display number.

## Future Plans

1. **Better method of f-stop exposure strip timing:** remove need to keep hitting the Up/Down buttons to move to the next test strip's time.
2. **Add functionality specifically for paper processing:** some people use the timer for paper development using the memories, but a better option is to implement a simple process timer with named processes that can be saved and loaded.

Always happy to hear suggestions, but please note that this is a hobby project, so no promises about timescales...

## Basic Operation

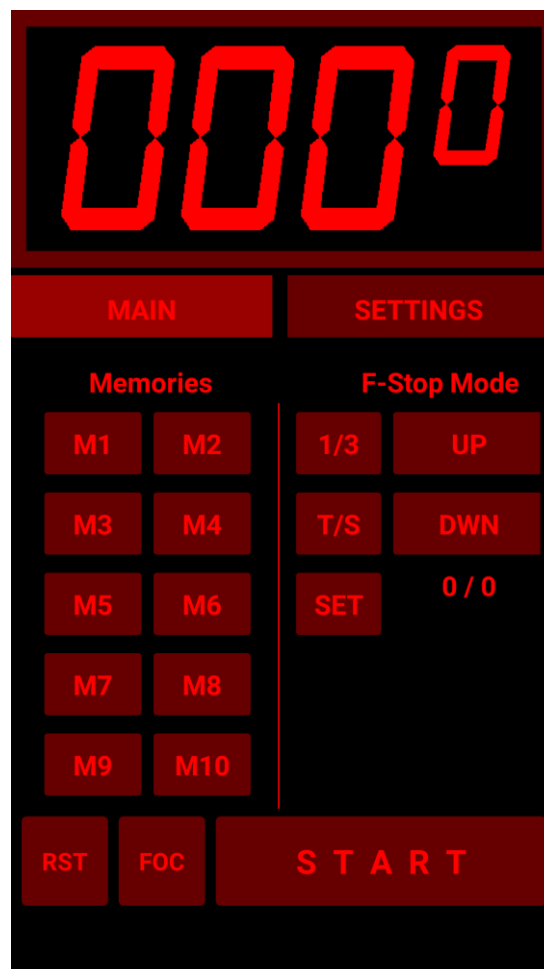
The basic operating method is much the same as any traditional digital enlarger with the UI adapted for a touch-screen.

The timer is primarily intended for use with black and white photosensitive paper and uses different intensities of red to display the user interface. Be aware that most Android devices' LCD screen will leak light that can fog black and white paper even if the display looks deep red to the human eye so you are advised to use the phone's built-in screen brightness control to keep the brightness of the display as low as practical in the darkroom and avoid shining the display directly on photosensitive paper. If used for colour printing where the paper is sensitive to almost all colours of light, it may be necessary to shade the display with black card to ensure no light spillage. As always, test your materials and equipment for safe lighting conditions before performing any serious printing.

There are two screens on the timer: The *Main* screen is where all the timing operations are performed, and the *Settings* screen is where the various options are configured. Tapping on the tabs switches between the two screens.

See Appendix 1 for details of connecting to one of the external switching options, but for manual use set the "Lamp Controller" option to "**None**" on the Settings tab.

## Main Screen



Dealing with the Main screen first, the basic timer operation is to tap the top of bottom half of the required number to set the time and then hit the [START] button to commence the countdown to zero. The Focus button [FOC] can be used to manually toggle the lamp on and off.

### **[START]** Start/Stop

The Start button counts down to zero from the time shown on the timer display. Hitting the Start button (renamed to [STOP] for the duration of the count) before the time reaches zero will suspend the count.

If the timer is connected to external switching hardware, it will switch the enlarger lamp on at the beginning of the count and off again when the count reaches zero.

### **[RST]** Reset

The Reset button has two actions:

- Long press: resets the timer count to zero
- Short press: resets the timer count to the previous value used

### **[FOC]** Focus

The Focus button toggles between switching the enlarger lamp on and off. Requires linking to a smart socket to operate the enlarger lamp.

### **[M1] – [M10]** Memories

The ten memories have two actions each:

- Long press: stores the current time showing on the Display
- Short press: restores the previously saved time

Memory times are stored on the Android device, so you can store common times and retrieve them the next time you launch the app. Most of the values on the Settings screen are also saved automatically if changed. This is why the app asks for permission to access your phone's data when it is installed. Storing and retrieving settings in this way is the only access the app makes to the phone (other than using its WiFi connection to talk to the lamp-switching hardware).

Some users are employing the memories to hold standard times for paper processing. I have plans to expand the functionality of the timer to make this use more practical.

## F-Stop Mode

The previous instructions allow use of the App as a standard electronic enlarger timer - this section describes the use of the timer in the more advanced f-stop timing mode.

Experienced darkroom workers will be aware of the advantages of the f-stop method of working over the more common linear timing method, but a short description is available in Appendix 2 below which users new to this method may wish to read. There are also multiple descriptions and videos available online from many different sources that describe the concept in various levels of detail.

Operation:

The timer operates to the nearest  $1/10^{\text{th}}$  of a second, however, to maintain accuracy the internal calculations retain full mathematical floating-point precision between increments and decrements so do not suffer from rounding errors.

Note that to save space, the buttons used in this area of the screen are not labelled so will be referred to by the values showing in the screenshot below.



**[SET]** Set Base Time

This button takes the current time in the Seconds Display as the base time for subsequent f-stop calculations. For example, if you were creating a test print with test strips starting at 12 seconds, you would set the Seconds Display to 12 and press the [SET] button. The base-time part of the f-stop display (currently showing "000 / 0" in the screenshot) will be set to: "012 / 0".

Subsequent use of the Up/Down buttons will use this time as the base-time. This is always visible to help avoid any confusion about what your base-time is and how many times you have hit the up/down buttons.

**[UP] [DOWN]** Up/Down buttons

These buttons increment and decrement the base time by the value set on the F-Stop Adjust button - currently showing [1/3] (=  $1/3^{\text{rd}}$  of a stop) in the screenshot. The Count part of the base-time (currently showing "/ 0") will increment/decrement for each button press as appropriate. Hence, if the base time is 12 seconds and you have pressed the Up button four times, the f-stop display will show "012 / 4".

### [1/3] F-Stop Adjust

When the app is started, the f-stop value that will be used to adjust the timing will default to 1/3<sup>rd</sup> of a stop. Clicking on the f-stop adjust button will open a new screen where you can choose between several common fractional values (1, 1/2, 1/3, 1/4, 1/5, 1/8, 1/10):



Once selected, the select screen will close and the value displayed on the button will change to the new value to be used in subsequent calculations.

### [STR] Set f-stop calculation mode

There are two modes available when using the f-stop printing method. This button toggles between the two modes:

[ADD] Addition mode:

This is the simplest mode and simply adds or subtracts (using the Up or Down buttons) the F-stop Adjust value to the current time.

This mode is useful if you are finding a suitable time for dodging/burning etc or if you are using individually timed strips to find the required exposure time.

For example, using the values displayed in the screenshot above, if you had set the Base Time to 12 seconds (i.e. put 12 on the Seconds display and press the [SET] button) pressing the Up button would display 15.1s (12s + 1/3<sup>rd</sup> stop = 15.1s). You will see that the small display has changed to "012 / 1" indicating that the Seconds display is showing 12s + 1 x 1/3<sup>rd</sup> of a stop). Pressing the [START] button will now commence the 15.1s exposure.

Pressing the up button again will change the Seconds display to "019.0" and the small f-stop display to "012 / 2" (12s + 2 x 1/3<sup>rd</sup> stop). Pressing Up again will add a further 1/3<sup>rd</sup> of a stop to set the timer to "024.0" (i.e. one whole stop above 12s as we have added 3 lots of a 1/3<sup>rd</sup> of a stop to the base 12s time). The Down button will work in the same way but subtracting 1/3<sup>rd</sup> of a stop each time.

**[STR]** Test-strip mode:

This sets the Seconds Display to the difference between the previous time and the new time. For example, if the Base time is 12s, pressing the Up button will set the timer to 3.1 seconds

This mode facilitates the creation of exposure test-strips on a single sheet of paper with f-stop exposure changes between strips. The f-stop value (between 1 and 1/10<sup>th</sup> stop) to use is set by the F-stop Adjust button as previously described.

For example, to create a single sheet test-sheet you might perform the following: Note that this example is using the test-strip method where you first expose the whole test sheet paper to your base time and then progressively cover a section of the paper making additional exposures each time until the paper is fully covered.:

1. Decide on your minimum exposure – you probably have some idea of this from previous printing experience. For the sake of illustration, let's say that you expect the final exposure to be somewhere around 35 seconds for the print size, lens aperture, etc you want to use. You decide to start at 20s as you don't believe the final exposure will be less than 30s, but 20s gives a fair amount of wiggle-room just in case.
2. Use the F-stop Adjust button to set the f-stop increments you want each test-strip to increase by. Let us say you select 1/4 stop.
3. Set the mode button to [STR] as we are exposing multiple strips on a single sheet of paper.
4. Set 20 on the Seconds display and hit the [SET] button to set this as your base time. We will now make several exposures from this base time.
5. Press the [START] button to expose the whole sheet of paper for 20s. Wait for the lamp to switch off.
6. Press the [UP] button. The time will change to 3.8s and the small f-stop display to "020 / 1". Cover the first approx. 1/6<sup>th</sup> of the paper width and press the [START] button to make the 3.8s exposure. Wait for the lamp to switch off.
7. Press the [UP] button. The time will change to 4.5s and the small f-stop display to "020 / 2". Cover the second 1/6<sup>th</sup> of the paper width and press the [START] button to make the next exposure.
8. Press the [UP] button. The time will change to 5.4s and the small f-stop display to "020 / 3". Cover the third 1/6<sup>th</sup> of the paper and press the [START] button to make the next exposure.
10. Press the [UP] button. The time will change to 6.4s. Cover the next 1/6<sup>th</sup> of the paper width and press the [START] button.
11. Press the [UP] button. The time will change to 7.6s. Cover the next 1/6<sup>th</sup> of the paper width and press the [START] button.



12. That should have been the last strip to cover, so we are done. If you misguessulated the strip sizes and still have a lot of paper to cover, just continue in the same vein until you run out of paper...
13. Process the paper in your usual manner and look for the best exposure.

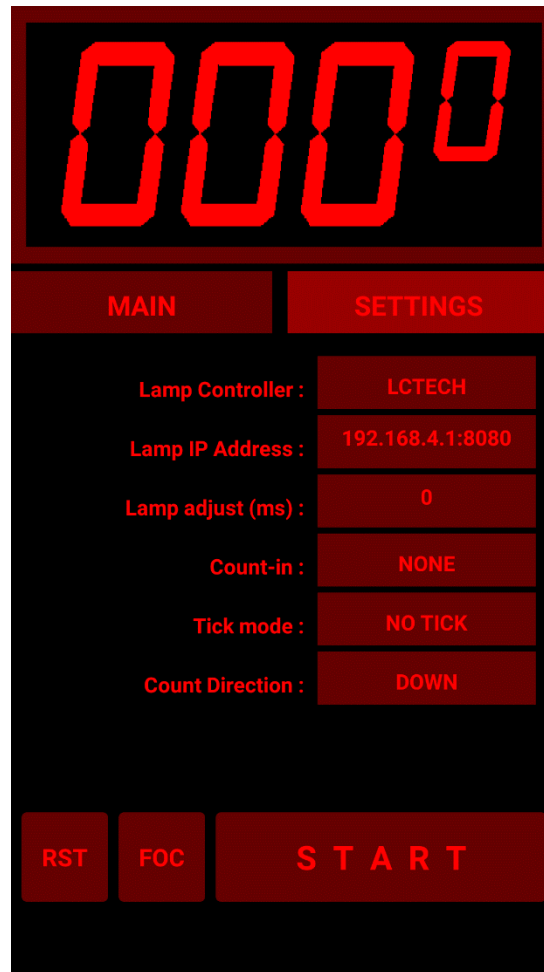
Tips:

In the above example where we decided to start at 20s, another method is to set your base-time to 35s, set the f-stop adjust value to  $\frac{1}{2}$  stop and select [ADD] mode. Now pressing the Down button twice will give you a new Base time at one stop below your original time and you can hit [SET] to use this as your new base time for your test strip (remember to change to [STR] mode and set your required f-stop adjust value before starting the strip). If you wanted to start at some other time than 1 stop below the expected time you simply choose the f-stop adjust value and the number of times you press the [DOWN] button accordingly.

If you have no idea what exposure you will need, set the F-stop adjust to  $\frac{1}{2}$  stop (or even 1 stop) and set the base-time to 3 seconds and use a quarter-sheet of paper with 6-8 strips as above to get within  $\frac{1}{2}$  stop of the required exposure and perform a second test strip around this time with smaller f-stop increments. For example, using a base time of 3 seconds and  $\frac{1}{2}$  stop increments, with 8 strips you can span 3s to 48s which should get you in the ball-park for a more precise test.

If your shortest timed strip is too dark, stop down the enlarger lamp a couple of stops and try again. If still too dark, you may have to introduce some neutral density filtration.

## Settings Screen



**Lamp Controller:** Selects between the Sonoff and LC Technologies hardware interfaces and None.

If you are not using either hardware interface, select **None** here (or **SONOFF**) because the **LCTECH** interface will cause Android error messages to pop up on the screen if you are not connected to the module via WiFi. **None** is selected by default when the timer is first installed but changes to this value are stored in your device's memory thereafter and loaded each time you restart the timer app.

**Lamp IP Address:** When used with the Sonoff hardware switch interface, this is the IP address that was assigned to the hardware during reprogramming.

This defaults to 10.0.0.1 but can be changed and the change will be preserved across restarts of the app. If the target Sonoff switch has been programmed to operate on another network port it can be changed here in the same way you would in a web browser address bar (e.g. *10.0.0.4:8080* to write to a Sonoff S20 using port 8080 on IP address 10.0.0.4 ).

I find it best to connect to the Sonoff or LCTech WiFi network access points before starting the timer app to ensure the timer app connects to the hardware correctly.

**Count-In:** You can optionally set a 3-second delay before the timer starts to count after hitting the [START] button.

This button switches between **3-SECONDS** and **NONE**.

**Tick mode:** The timer can optionally click on each second. There are three options:

**TICK:** The timer ticks every second. Can be useful when you are dodging/burning etc and don't want to glance at the timer – you can just count the seconds in your head.

**NO TICK:** timer does not make a ticking sound.

**LAST 5 SECS:** the timer will tick off the last 5 seconds of the count, but only if the count was for 10 seconds or more. Can be useful to alert you to the fact that the count has nearly finished if you are switching the lamp manually.

**Count direction:** Toggles between Counting UP and counting DOWN. Count = UP can be useful for continuous ticking to provide a metronome effect, but count = DOWN is the normal mode. The timer always starts up in Count direction = DOWN.

**Lamp Adjust:** When a lamp switches on it takes some time to reach full brightness. When a lamp switches off, it takes some time to decay to full darkness. These times and the precise ramping up and down of the brightness will differ and hence the amount of light that reaches the photographic paper are rarely the same for ON and OFF actions of a lamp.

Often the differences are too small to be detectable in our prints. Other times, they are larger and can accumulate over several additive exposures to affect our prints. This depends on the type of lamp used and if, for example, there is a transformer or voltage stabiliser circuitry in the path. For example, if we have a difference equivalent to  $1/3^{\text{rd}}$  of a second less exposure, four 3 second exposures used on a test-strip will not be the same as a single 12 second exposure – the 4 x 3 second exposures will be only 10.67 seconds - short by  $1/6^{\text{th}}$  of a stop. A  $1/6^{\text{th}}$  stop difference may be easily noticeable in the final print. Smaller differences may not be so noticeable but can still produce more subtle effects.

This setting allows you to set an adjustment to the first second of the count to offset this difference. The times entered here are in milliseconds (so 500 entered is 0.5 seconds, 250 is 0.250 seconds etc). You can also use negative numbers (e.g. -100 = minus 0.1 seconds). A positive number adds that number of milliseconds to the count and a negative value subtracts that number of milliseconds.

This value is automatically saved to the Android device whenever it is changed, and it is reloaded when the app is restarted.

You will need to obtain the correct value to insert here by trial and error. For example, make a single 20s exposure of a section of a negative with good highlight and shadow detail and then take ten 2-second exposures on a separate piece of paper without changing anything in the setup. There is no need to use whole sheets – a small piece is fine as long as it shows the good highlight and shadow detail from the negative. Mark which is which, process them together and observe any

difference in the exposure of the two sheets once the paper has dried (use resin-coated paper for this test for speed but if you have a choice of different papers, use the most sensitive). If the 10x2-second exposures sheet is lighter than the single 20 second exposure sheet then you need to make a +ve adjustment to increase the final second of the timer, and if darker you will need to make a -ve adjustment. It may take several iterations to find a suitable value.

## **Appendix 1: Connecting the App to the Lamp Switching Hardware**

You will need to configure the app to connect to the device by selecting the appropriate hardware interface.

Note that trying to connect to the LCTECH device will cause an error if your Android device is not already connected to the relay module. Note that the timer is designed to ONLY connect to this specific LC Technology relay board.

### **Sonoff S20 (or S26) Smart Socket**

You will need to disconnect from your home network and connect to the separate network used by the Sonoff switch. The WiFi Network is called "FADUTimer". You will need to connect your phone/tablet to this network in the usual way. The password has a default value in the code (fadu1234) but it is recommended that you change this to something more relevant to yourself.

Configuring the S20 as an Access Point (AP) with its own WiFi network and embedded web server in this way has the advantage that you do not need to hunt for whatever IP Address your WiFi router has assigned to the unit if you were to connect it to your existing home network and also removes the device from the public Internet so it cannot be accessed by miscreants there. However, your neighbours will be able to see the FADUTimer network, which is why you should change the password just in case you have smarter than average neighbours...

#### **To summarise:**

1. Plug the reprogrammed Sonoff S20 into a power outlet and plug your enlarger into the Sonoff.
2. Connect your Android device to the FADUTimer network
3. Open the FADU Enlarger Timer app
4. On the Settings tab, select "SONOFF" and insert the Lamp IP Address (10.0.0.1 by default).
5. Press the Focus [FOC] button and the enlarger lamp should switch on.
6. Pressing [FOC] again should switch the enlarger lamp off.

If you find the lamp is sluggish to respond, put the Android device into Aeroplane mode. This prevents the phone from doing a lot of background tasks which can add unwanted delays. You will need to re-enable WiFi and connect to the FADUTimer network again after switching to Aeroplane mode.

## **LC TECH relay board**

***Do not connect this board to mains voltages. Use it only on the low-voltage side of a transformer/voltage regulator.***

This board does not require programming as it can connect directly to your Android Device via WiFi.

The board has two relays that are both switched by the FADU app. There is a 1 second switching delay included between Relay 1 and Relay 2 so if you want your enlarger lamp to switch with a one second delay, connect it to Relay 2.

You will need to disconnect your phone from your home network and connect to the separate network used by the IT Tech relay board. The WiFi Network is called "AI-THINKER\_XXXXXX" where "XXXXXX" is a random 6-digit hexadecimal number (this is quite useful as it allows you to have more than one enlarger equipped with this board in the same room and control them via different Android devices).

You will need to connect your Android device to this network in the usual way before switching the timer to use the device or you will get a pop-up error message. Depending on the version of Android you have, the error message may disappear after a couple of seconds, or may require you to tap anywhere else on the screen to dismiss it without closing the timer app.

Please see Appendix 5 for suggested hardware connections.

### **To summarise:**

1. Supply power to the relay board
2. Connect your Android device to the "AI-Thinker\_XXXXXX" network
3. Open the FADU Enlarger Timer app
4. Press the Focus [FOC] button and the enlarger lamp should switch on
5. Pressing [FOC] again should switch the enlarger lamp off

If you find the lamp is sluggish to respond, put the Android device into Aeroplane mode. This prevents the phone from doing a lot of background tasks which can add unwanted delays. You will need to re-enable WiFi and connect to the AI-Thinker network again after switching to Aeroplane mode.

## Appendix 2: F-Stop Timing

The f-stop printing method is a way of finding the correct exposure for a print by using the inherent properties of photosensitive photographic paper - just as your camera uses the same properties for film when you adjust the f-stop or exposure time on your lens or camera body.

Making a test strip using equal times for each strip results in unbalanced exposure strips across the sheet of paper. For example, a sequence of 5 second strips will result in the second strip (10s) being twice the exposure of the first (5s), the third strip (15s) is 1.5x the exposure of the second, the fourth (20s) is 1.3 the exposure of the third and the fifth (25s) is 1.25 the exposure of the fourth, and so on.

You can see that as the strips progress, we have much greater exposure difference at one end than we do at the other. With f-stop printing this difference is evened out to obtain strips of equal increase in exposure density across the sheet.

Using the f-stop method and using the original 5s starting exposure and incrementing by 1/2 a stop for each subsequent strip we get: 5s, 7s, 10s, 14s, 20s which corresponds to equal increases in exposure for each strip, making the determination of the correct exposure much easier and accurate.

### Further reading:

Searching the web for "f/stop printing" will find many articles, forum posts and videos describing the method in more detail and the worked example in the manual above shows one way of using the FADU timer to generate test strips and to adjust a single exposure time by a selected f-stop.

## Appendix 3: Sonoff S20 Programming

Anyone familiar with using the Arduino system should not find this too difficult – especially if they have used an FTDI adaptor before and have used a soldering iron.

Note that once programmed the original firmware for the Sonoff S20 will be lost without any possibility of recovering it - it will never work as originally supplied by the manufacturer again.

What you need:

1. The Arduino IDE software: download from <https://www.arduino.cc/en/Main/Software> and install.
2. Sonoff S20 Wi-Fi enabled smart socket.
3. FTDI adaptor: 3.3v version.
4. USB cable for the FTDI board: make sure it can synch – some cheap cables are sold for phone charging only and do not have the necessary signal lines connected.
5. Dupont type leads: female <--> female.
6. Pin Header: 4 pins of 0.1 inch pitch.



All these are readily available on eBay or Amazon – other suppliers are available. Make sure you get the correct S20 smart socket for your country’s power socket type. One reason for choosing the Sonoff brand was because they are available in several regional variations – the above shows the UK type.

The FTDI adaptor (named after the manufacturer of the original and most popular chip used in these boards: “FTDI” <https://www.ftdichip.com> ) is more generally called a “USB to serial port adaptor”.



The FTDI adaptor used for this project MUST be 3.3V compatible. Many boards are switchable between 5V and 3.3V via a jumper or switch but some are 5V only – do not buy one of those. Make sure you configure it for 3.3V – applying 5V to the S20 will kill it stone dead.

If you prefer, you can skip using the header pins and temporarily solder flying leads to the S20 directly.

### **Prepare the Sonoff S20**

To remove the circuit board from the S20, remove the three screws from the back of the case and the top part will pull away. If you want to temporarily solder the leads directly to the board, this is as far as you need to go as you can CAEFULLY solder the leads to the board now, but if you want to solder the header on the board (recommended) then you need to remove the two screws holding the board to the base (the heads on these screws are easily damaged – make sure you use the correct size screwdriver).

The board on the ones I have had are very stiff and need GENTLY levering up next to the screw holes after removing the screws – use something plastic if possible. Removing the top part of the socket also makes access to the header holes easier but watch out for the spring holding the pin shutters in UK type sockets – it easily pings out.



The connections to the FTDI adaptor are visible to the lower left of the image above, as is the push switch (centre) that we will also need during programming. The pins are as follows per the image above from top to bottom:

- O Gnd
- O Tx
- O Rx
- O < Vcc

These need to be connected to the corresponding pins on the FTDI adaptor:

### **S20 to FTDI Pins**

Gnd to Gnd (Ground 0V)  
Tx to Rx (serial data from S20 to FTDI)  
Rx to Tx (serial data from FTDI to S20)  
Vcc to Vcc **(3.3V ONLY)**

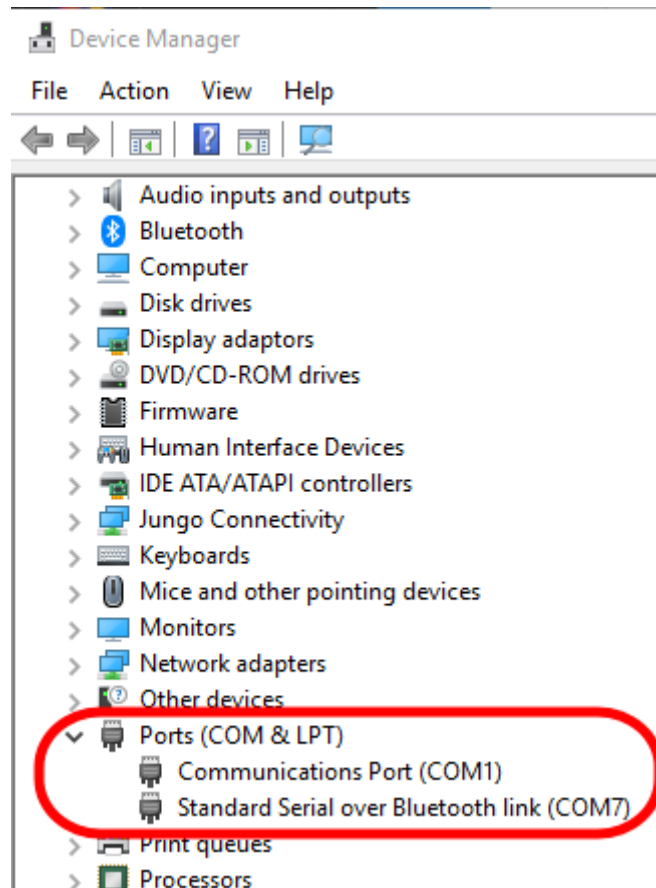
Other pins on the FTDI adaptor are left unconnected.

*Tip: if possible, get an FTDI adaptor that uses an actual FTDI chip. As mentioned, “FTDI” is actually the name of a chip manufacturer who make the USB-serial chip. Other companies make similar chips, but you may have to search the web for the drivers for non-FTDI chips. FTDI manufactured chips are automatically detected by Windows so the drivers are installed automatically, saving this extra step if you are not familiar with these things. Non-FTDI chip boards will still work fine – it is just an extra faff to step through to load the drivers.*

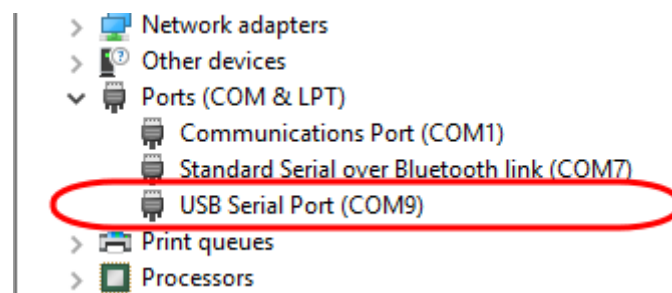
### **Prepare the FTDI adaptor**

You will need a USB cable to connect the FTDI board to your computer. There are a few that plug into a USB port directly, but they are less common. Once plugged in and the drivers installed the FTDI adaptor will look like a serial port to your computer (this is its job – to connect to external hardware as a serial port to allow communications between the computer and the external device).

I can't help with Macs and Linux based PCs (but there is a lot of info on this online) but on a Windows 10 PC, to find out which port number it is on, open the Windows Device Manager (hit the Windows key and start typing “device manager” to find the Device Manager). Display the Ports section in Device Manager and it should look something like below before plugging the FTDI board in:



After plugging the FTDI board into your PC, and assuming your board uses a FTDI chip, the PC should load the necessary drivers and your board will be recognised as a serial port as below - probably with a different port number – this one has been assigned to Serial Communications Port 9 by Windows. If your board does not use an FTDI manufactured chip, search online and load the drivers separately if necessary:



When you can see this extra Com port, your FTDI board has been recognised and is ready to use. If after refreshing the Device Manager it is still not recognised you will need to troubleshoot, in which case google is your friend – just search for: FTDI Arduino problem.

## Prepare the Arduino IDE

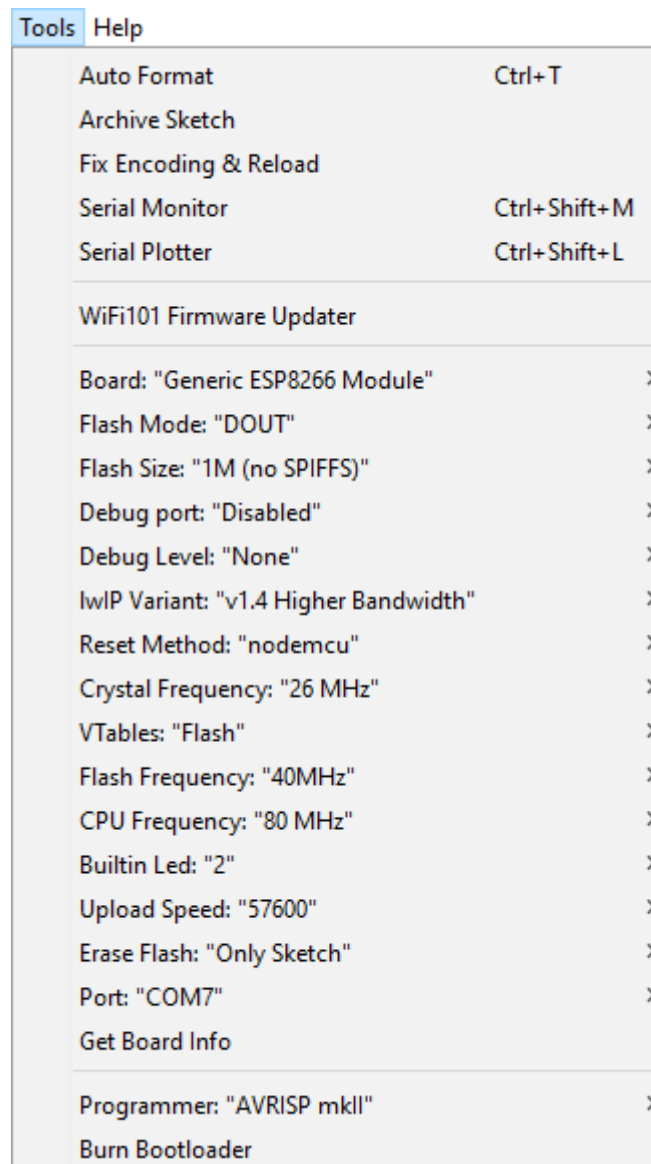
By default, the Arduino IDE does not know about the ESP8266 Wi-Fi chip used by the S20 so we need to load a Board Manager library so we can program it.

Run the Arduino software and from the menu bar select **Preferences**. In the **Additional Boards Manager URLs:** box enter: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) and click **OK**.

Now we have told the Board Manager where to look for new board configurations, open the Boards Manager itself (Tools menu => Boards Manager). In the window that pops up, scroll past any other items and select the **ESP8266 by ESP8266 Community** item. Click Install and a number of ESP based board configurations will be added to your Arduino software.

Open the Tools menu again and by scrolling down you should now see an entry for **Generic ESP Module** – select that.

Also note that some of the other settings on this menu may need changing. You should set it to the settings shown below (except the **Port** setting which will need setting to the port your FTDI adaptor has been mapped to):



In the Arduino IDE code window, delete the default template code and paste the new code for the FADU Timer. Save this to somewhere safe - installing the Arduino software will have created a default location, so you may as well accept that.

Before plugging in the FTDI board and trying to program the S20, we can do a quick test that all is well on the software side by compiling the code in the Arduino IDE: open the Sketch menu and select Verify/Compile (or hit CTRL+R). You should see some informational messages in the bottom part of the window and (hopefully) a list of the ESP8266 chip's resources used by the program and no error messages.

Once the code has compiled OK we are ready to program the S20.

### Programming the S20

1. **MAKE SURE THE S20 IS NOT PLUGGED IN TO THE MAINS.** The FTDI adaptor will supply the power to program the S20 for programming and if the S20 is plugged into the mains, you will not only blow the FTDI board but will possibly fry your PC and/or yourself too. So don't.
2. If you are soldering the header pins to the S20, do so.
3. Load the new firmware code into the Arduino IDE and compile it to make sure all is well.
4. Unplug the FTDI board from your computer – we do not want it plugged in yet.
5. Connect the S20 and FTDI boards together using the flying leads. Make sure you have the wiring correct – especially the GND and VCC power leads – get these wrong and the S20 is toast. The other two leads may be swapped without damage but the power leads must be connected correctly (and VCC must be 3.3V - I may have mentioned that before). Double-check these connections.
6. Make yourself a cup of tea/coffee.
7. Check them again...
8. Hold down the switch and keeping the switch pressed, plug the FTDI board into your computer. This supplies power to the S20 while the switch is pressed, putting the ESP8266 chip into programming mode. After a second or two release the switch. You should see the LED flashing. The board is now ready for programming.
9. In the Arduino IDE, open the sketch menu and select **Upload** (assuming you have already compiled the code). If not already compiled, hit the right-pointing arrow below the menu bar to first compile and then upload the new firmware.
10. Once the new firmware has been uploaded, remove the cable from the Sonoff's header and reassemble the device.
11. Plug the Sonoff into a mains socket and use the timer app [FOC] button to see if you can hear the device's relay switching.

## The new firmware:

```
/** Create a WiFi access point and provide a web server on it */

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

/** Use AP IP Addr: 10,0,0,1 */
IPAddress local_IP(10,0,0,1);
IPAddress gateway(10,0,0,1);
IPAddress subnet(255,0,0,0);

/** AP Credentials - feel free to change */
const char *ssid = "FADUTimer";
const char *password = "fadul234"; // Reccommend changing this - must be at least 8 characters

/** Start the web sever */
ESP8266WebServer server(80);

/** routine to send on connect to web server */
void handleRoot() {
  server.send(200, "text/html", "<h1>You are connected</h1>");
}

/** sonoff stuff */
int gpio13Led = 13;
int gpio12Relay = 12;

/**** Initialise */
void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();

  Serial.print("Configuring access point...");
  Serial.println(WiFi.softAPConfig(local_IP, gateway, subnet) ? "Ready" : "Failed!");

  Serial.print("Setting soft-AP SSID & Pwd... ");
  Serial.println(WiFi.softAP(ssid, password) ? "Ready" : "Failed!");

  /** Debug config OK */
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/", handleRoot);

  /** sonoff stuff */
  pinMode(gpio13Led, OUTPUT);
  digitalWrite(gpio13Led, HIGH);

  pinMode(gpio12Relay, OUTPUT);
  digitalWrite(gpio12Relay, LOW);

/** Routines to handle API calls - really complicated...?! */

/* "/on" = switch enlarger lamp on */
server.on("/on", [](){
  digitalWrite(gpio13Led, LOW);
  digitalWrite(gpio12Relay, HIGH);
  delay(500);
});

/* "/off" = switch enlarger lamp off */
server.on("/off", [](){
  digitalWrite(gpio13Led, HIGH);
  digitalWrite(gpio12Relay, LOW);
  delay(500);
});
```

```
server.begin();  
Serial.println("HTTP server started");  
  
}  
  
void loop() {  
  server.handleClient();  
}
```

#### References:

<https://medium.com/@jeffreyroshan/fl...de-402e5a2f77b> - uses a different Sonoff WiFi switch, but is essentially the same process.

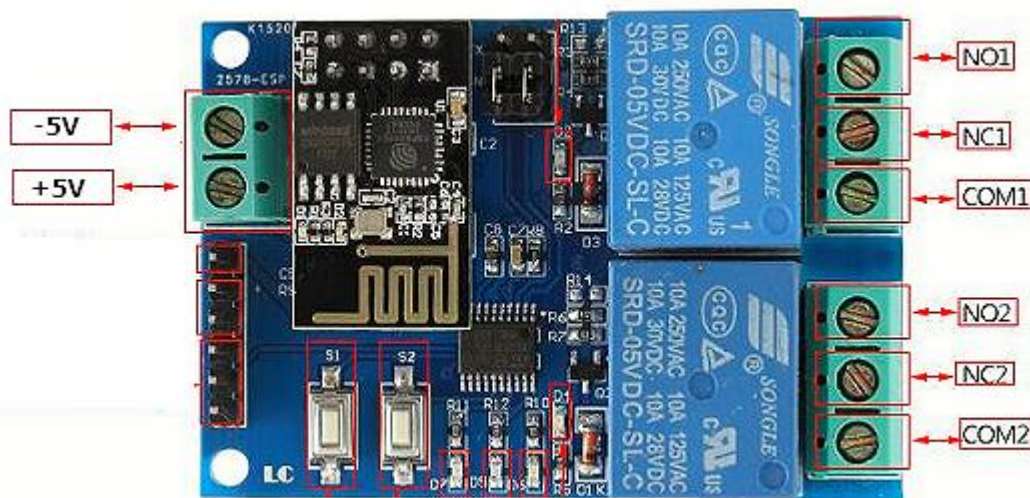


## Appendix 5: LC Technologies Relay Board

\*\*\* DO NOT CONNECT THIS BOARD TO THE MAINS \*\*\*

This is a bare board so will require a power supply (regulated 5 volts dc), placing in a box, and connecting to the low-voltage side of your enlarger's transformer/voltage regulator. You can leave it disconnected from the enlarger until you have tested the board for correct operation.

On the plus side, it requires less electronics knowledge to configure for use – reasonable DIY skills should suffice. A relay is simply an electrically operated switch. Powering on the relay will cause the internal switch to move from connecting the Common terminal from the Normally Closed terminal to the Normally Open terminal.



The connections of interests are:

+5V : The positive side of the 5 volt power supply

-5V : The negative (ground/earth) side of the 5 volt power supply

COM1, COM2 : The common terminal of each relay

NC1, NC2 : The Normally Closed terminal of each relay

NO1, NO2 : The Normally Open terminal of each relay.

The NO terminals are not connected to the COM terminal of the respective relay when the board is unpowered and when initially powered on. When the FADU app tells the lamp to switch ON, the relays energise, and this terminal will connect to the COM terminal.

The NC terminals are connected to the COM terminal of the respective relay when the board is unpowered and when initially powered on. When the FADU app tells the lamp to switch OFF, the relays are de-energised, and this terminal is connected from the COM terminal.

Hence, to drive the lamp on and off we need to break into one of the leads going to the enlarger lamp from the transformer unit and connect them to COM1 and NO1. It does not matter which way around they are connected.

A suitable regulated 5V dc power supply is readily available from the usual online suppliers. Note that this MUST be 5 volts – any lower voltage will likely not work, and a higher voltage will likely kill the board. Many available power supplies are adjustable between 3 and 12 volts – make sure it is switched to 5V before connecting it and that the polarity is correct.

The image below shows the board placed in a case. A 3.5mm DC power socket has been used to take the power from the power supply and the end of the case has been cut out to provide access to the relay terminals. I have simply cut into one of the low-voltage wires from the enlarger's voltage stabiliser and screwed the ends into Relay-1's COM1 and NO1 terminals – a more elegant solution would use additional connectors.



Driving my LPL C7700:w

