FADU Enlarger Timer

[Version 1.5 02-Mar-2021]

Introduction

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

Of special note for version 1.5 is the ability to use Sonoff Basic R3 and Mini devices without the need to reprogram the firmware. See the Release Notes below and the Appendix 1 section on using the Sonoff devices in DIY mode. The manual also now details ways to reprogram the firmware of these devices using OTA (Over The Air) updates.

The timer can be used manually with an on/off switch or can be connected to a Wi-Fi enabled Smart Socket or Relay module to switch the enlarger lamp. There are currently four options to drive external hardware in this way:

1) A Sonoff S20 or S26 smart-socket which can be plugged into the mains directly with the enlarger plugged into the Sonoff. To use this, you will need to reprogram the firmware on the Sonoff which is not trivial. You can only switch mains voltages with these devices. Full details in the Appendix.

2) A Sonoff Basic or Mini Wi-Fi mains relay switch. Using one of these devices in DIY mode is my recommended method if you do not wish to reprogram a Sonoff device - this only requires simple DIY skills to wire them and place in a suitable box. You can only switch mains voltage with these devices.

   The mains power is connected to terminals on the input side and the mains connection to the enlarger is connected to the output terminals. There are three modes available to use these units, two of which use the DIY mode API available in newer (R3) versions of these units which requires no reprogramming and a third way that requires reprogramming the units as for the Sonoff S20/S26. The Mini and Basic models require placing in a box as there is insufficient (or no) cable retention, exposed terminals (Mini) and no earth/ground continuity. See Appendix 1 for details of the DIY modes.

3) A widely available LC Technologies relay module board which can be used to switch the low-voltage side of an enlarger's transformer/voltage regulator. This needs a 5V power supply and a box to put it in – no programming is necessary.

4) Make your own. The source code for the app is available on Github and can be edited in MIT App Inventor 2 to suit your own design of hardware. The code for the ESP8266 Wi-Fi chip used in the Sonoff devices is included in this manual. Many projects exist online using the ever-popular ESP8266.

Switching the enlarger lamp

I have been unable to find any commercial Wi-Fi 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 purposes due to the inherent unpredictable time delays introduced by this connection method.

This means that we may need to reprogram the firmware on an existing device, however see Appendix 1 for details on using the Sonoff Basic R3 and Mini devices without reprogramming. The S20 and S26 Wi-Fi smart sockets are widely available in local mains voltage and socket type options and have an online community of makers and hobbyists who customise the devices with new firmware. Likewise, the Sonoff Basic and Mini models are widely available.

It is not excessively difficult for the technically minded to reprogram these devices. The newer DIY R3 versions of the Basic and Mini can be programmed via an OTA (Over The Air) process that avoids the need to use additional hardware but this does have a few hoops to pass though as Sonoff have recently changed the firmware on these devices to 3.5 (and above) which has made flashing them with new firmware more complicated. Only the Basic R3 and Mini units can be flashed via OTA, the S20/S26 etc do not have this ability (as of February 2021). This process is described later in this manual.

Alternately, they can be programmed using external hardware but that does require soldering a connector to existing pads on the circuit board and connecting your computer to the socket via a USB-to-Serial-port (FTDI) adaptor in order to download the code from the Arduino IDE to the Sonoff device. If you are at all familiar with programming Arduino system boards, this should cause few problems for you.

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

The Sonoff S20 is being replaced by the Sonoff S26 by the manufacturer which requires a slightly different method to reprogram as it has no connector pads so will require flying-leads to be soldered to the pcb from the FTDI adaptor, but it uses the same code.

As mentioned, the Sonoff Basic R3 and Mini models can be used (with some limitations explained in Appendix 1) without the need to reprogram them.

An alternative option if you wish to switch the low-voltage side of your enlarger setup is to use a Wi-Fi relay board from LC Technologies which is widely available. This does not require reprogramming so will work directly with the darkroom timer app without software modification, but it is an unboxed circuit board module so will need boxing and a suitable 5V regulated power supply.

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 very small (approx. 1mm) and this renders it unsuitable for use with high voltages on the terminals. This board should only be used on the low-voltage side of a transformer/stabiliser as is usually supplied with colour enlargers. Additionally, lamps greater than 150 watts at 24 volts (or 300w at 12V) should not be used without soldering additional thick copper wires between the relay contacts and the input/output terminals to increase the load carrying capabilities of the pcb tracks. See the Appendices for details.

I bought two LC Tech modules from Banggood. I also bought one from a random eBay seller which also worked OK but because there are so many no-name copies out there, there are no guarantees that they will have the same firmware installed.

A further alternative is to roll your own. There are many designs for relay switchers using the ESP8255 and the Arduino system online.

If you are unfamiliar with projects using mains voltages, please take very great care and obtain advice from someone you trust who has this experience.

It is not just electrocution that is the worry: there are failure modes that can result in a fire, so always disconnect home-made mains-powered projects at the wall socket when unattended.

Release Notes

Version 1.5.0:

(1) Added [PULSE R3] mode: uses a Sonoff Basic R3 or Mini device in the "Inching" mode of the R3 DIY mode API (http://developers.sonoff.tech/basicr3-rfr3-mini-http-api.html) to send fixed duration lamp-on commands to the Basic R3 or Mini in DIY mode. See Appendix 1 for details.

(2) Added [BASIC R3] mode: uses a Sonoff Basic R3 or Mini device in DIY mode. The downside of this mode is that all the lamp-on and lamp-off signals go via your Wi-Fi router which adds random delays of typically 5 - 100ms on my network - and occasionally much more. See Appendix 1 for details.

(3) Added the ability to use the camera's flash LED. Can be used for flashing paper if you add a diffuser over the LED and use the Delay setting to give you time to raise it to the right height after setting the time. Light activated relay boards do exist but keeping the bright light out of the rest of the darkroom may prove difficult. Could also be useful for testing or demo purposes.

(4) Added individual volume controls for the various beeps and buzzes (scroll down the SETTINGS tab if you do not see them). A bit tricky to use due to being small (I could find no way to enlarge them in App Inventor) - I find that touching the slider button for a second before sliding it is helpful. The settings are reloaded the next time the app is launched.

(5) You no longer need to use the [SET] button on the STRIPS tab. The strip's times automatically update when changing the base time or opening the tab. The [SET] button is retained although it is now believed to be redundant.

Version 1.4:

1. Single strip-mode [SINGLE]: added for when you are testing with individual strips of paper instead of making multiple exposures across a larger sheet of paper. Each strip's time in this mode is for the whole f-stop calculated exposure.

2. The Memories now display their contents. This is made possible by reducing the space taken by the F-Stop timing section now the SINGLE strip-mode has made it largely redundant. The Main tab's F-stop timing section is currently retained for people who want to generate f-stop times manually - although the Strips tab is better for this as it will list multiple times at once. This old functionality may be removed in a future version.

3. The display button text and text boxes etc now scale correctly for tablets. I cannot find a way to scale the lists that open in a new screen using AI2, but other than that it appears to scale well for larger screens. Tested on a 7-inch and a 10-inch tablet.

4. Added info on the Sonoff Basic Wi-Fi switch.

Version 1.3.0:

1. STRIPS mode tab added: This mode allows the easy creation of f-stop incremented test strips. Three methods are included - see Strips Mode chapter below for details.

a. Continuous mode [CONT]: intended for users with no lamp switching hardware. Sounds a beep 0.5 seconds before the end of each strip's time before starting the next strip time.
b. Auto mode [AUTO] : pauses the timer and switches off the lamp between each strip for a settable delay time.
c. Manual mode [MANUAL] : pauses the timer and switches off the lamp between each strip until the user taps [START] to move to the next strip.

2. Auto-Reset option added on the Settings tab. If set ON, resets the timer at the end of the exposure time to the previous time.

3. Delay time is now variable on the Settings tab from 0 to 10 seconds. An enhanced form of the previous "Count-in" setting.

4. About screen added to provide some basic info and links.

Version 1.2.0:

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. [*obsolete in 1.3.0 – now a link to GitHub on the About screen*]

3. Wi-Fi Relay Driver: Can now optionally drive an LC Technologies Wi-Fi relay module for low-voltage use. No reprograming 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. [*upgraded to 0-10s selection in 1.3.0*]

Version 1.1.0:

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

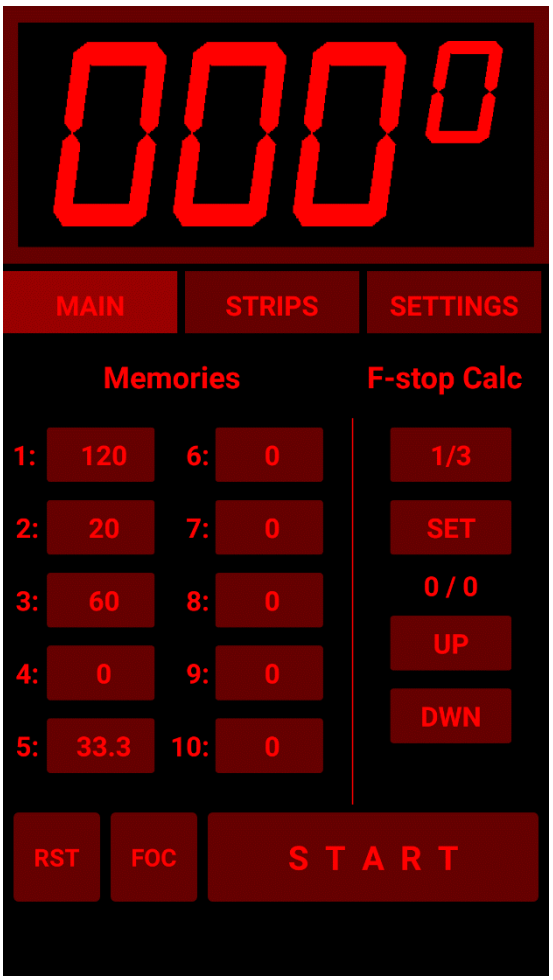
Main Tab

The basic operating method is much the same as any traditional digital enlarger timer, but with the user interface adapted for a touch-screen.

The timer is primarily intended for use with black and white photosensitive paper using different intensities of red to display the UI.

Be aware that most LCD screens use a backlight that can fog photographic paper even if the display looks deep red and black to the human eye. 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 when making changes to your darkroom lighting, test your materials and equipment for safe lighting conditions before performing any serious printing.

Covering the screen with "Rubylith" or similar transparent deep-red plastic film will allow a brighter display without danger of fogging B&W paper.



There are three tabs on the timer: The ***Main*** tab is where basic timing operations are performed. The ***Strips*** tab is for creating test-strips using f-stop timing techniques. The ***Settings*** tab is where the various options are configured.

See Appendix 1 for details of connecting to one of the external lamp switching options but for now, set the *Lamp Controller:* option to **[NONE]** or **[FLASH]** on the Settings tab.

Basic Controls

The basic timer operation is to tap the top or bottom half of the required number to set the time and then hit the [START] button to commence the countdown to zero.

[START] Start/Stop

The Start button starts the countdown to zero from the time set on the timer display.
Hitting the Start button (renamed to [STOP] for the duration of the exposure time) before the time reaches zero will suspend the countdown.

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

[RST] Reset

The Reset button has two actions:

Long press: resets the timer to zero
Short tap: resets the timer to the previous value

[FOC] Focus

The Focus button is used to manually toggle the lamp on and off if you have lamp switching hardware connected.

[M1] – [M10] Memories

The ten memories have two actions each:

Long press: stores the current time showing on the display
Short tap: restores the previously saved time

Memory times are stored on your 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 tab 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 Wi-Fi connection to talk to the lamp-switching hardware. Note that some recent versions of Android also require location permissions to use Wi-Fi - your phone will ask if it requires it - this information is not accessed by the timer app code.

F-Stop Calculation

This mode on the Main tab is now effectively made redundant by the more useful Strips tab modes but remains available for legacy reasons. This section may be removed in a future version if the space is required for new features.

This section describes the use of the timer in the f-stop timing mode. Experienced darkroom workers will be aware of the advantages of the f-stop method of working compared to the more common linear strip 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



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

[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 1/3rd 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, 1/6, 1/8, 1/10, 1/12).

Once selected, the screen will close, and the value displayed on the button will change to the new value.

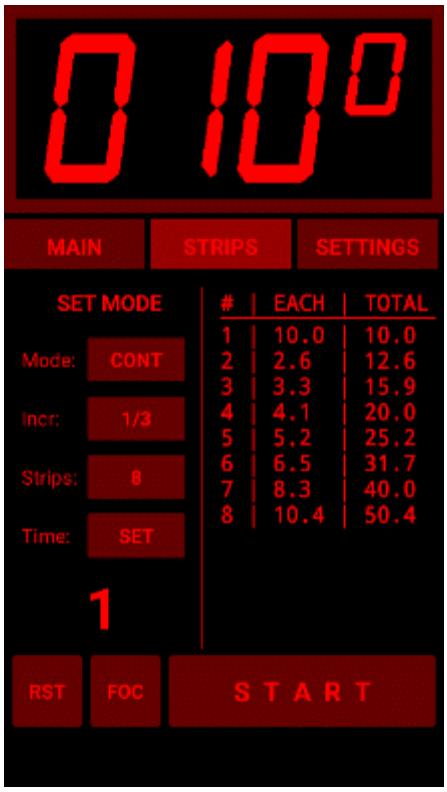
[SET] Set Base Time

This button takes the current time showing on the display as the base time for subsequent f-stop calculations. For example, if you set the timer's display to [012.0] and tap the **[SET]** button. The base-time part of the f-stop display (currently showing ***0 / 0*** in the screenshot) will be set to: ***12 / 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]** (= 1/3rd of a stop) in the screenshot. The Count part of the base-time (currently showing **/ 0**) will increment or decrement for each button tap as appropriate. Hence, if the base time is 12 seconds and you have tapped the UP button four times, the f-stop display will show **12 / 4**.

Strips Tab



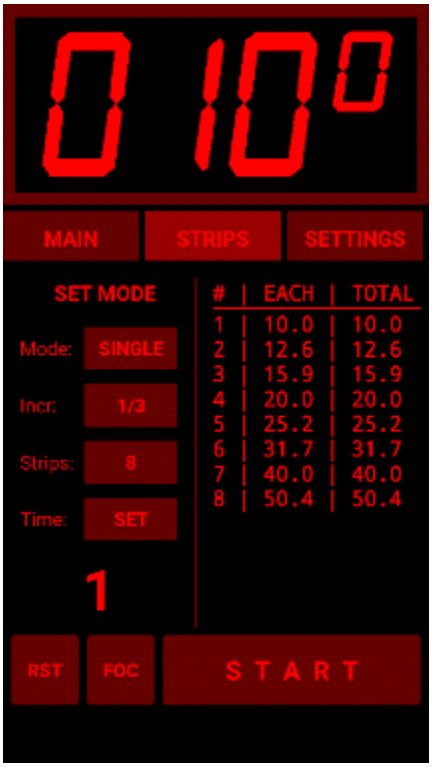
This tab allows the easy creation of f-stop incremented test strips. Four methods are provided, one intended for users with no lamp switching hardware and three for those who do have Wi-Fi connected lamp switching hardware. The test strip method used is where you first expose the whole sheet to the Base time (i.e. the first strip's time) and then cover each subsequent strip for the allocated time.

When setting the base time for the strips or changing one of the other parameters after setting the base time, the list on the right side of the tab is filled with the exposure time for each strip followed by the total exposure time that strip will receive once the timing has completed.

From this display we can see the time for the range of exposure times we will make. If we do not like what we see, we can change the parameters again and see the results instantly. For example, setting 1/3rd of a stop increments, 8 strips and a base-time of 10 seconds shows us the display on the left.

1) Continuous mode [CONT]: This mode is most useful for users with no lamp switching hardware. It provides a short beep 0.5 seconds before the end of each strip's time and then again at the end of each strip's time. It then continues counting down the next strip's time without pausing. Strip times under a couple of seconds will require careful attention from the user but the time for each strip is displayed so short times should not come as a surprise.

2) Auto mode [AUTO]: pauses the timer and switches off the lamp between each strip for a settable pause time before continuing with the next strip to give the user time to move the masking card to the next strip. The pause time defaults to 1 second but can be increased using the ***Settings*** tab's Delay value.



3) Single mode [SINGLE] : used when making exposures on individual single strips of paper, or when taking a sequence of test exposures from a single part of the negative. Each test strip time is for the complete f-stop calculated time - not for the incremental time as used when making standard test strips across a sheet of paper (see image on the right). Often used to fine-tune an important detailed part of the negative. Operates like Manual mode, stopping at the end of each strip time to allow the printer to prepare for the next test strip exposure.

4) Manual mode [MANUAL] : pauses the timer and switches off the lamp between each strip until the user taps [START] again to commence the timer for the next strip. Useful if you

do not want to be rushed between each strip as the AUTO mode will start the next strip whether you are ready or not once the pause time has been met.

Controls:

Mode: Selects between CONT, AUTO, SINGLE and MANUAL modes as described above.

Incr: Sets the f-stop increment for each strip.

Strips: Sets the number of strips that you wish to use across the test sheet of paper (but see the Tips section below).

SET: Sets the currently displayed timer display value as the base-time for the test strips. This will be the time for the first strip and all subsequent strip times are based on this and the f-stop increment value. The display is automatically updated if you change the Mode, Incr, or Strips values. From version 1.5.0 this also now updates whenever the STRIPS tab is opened and when the time display is manually adjusted.

Paper Exposure Process

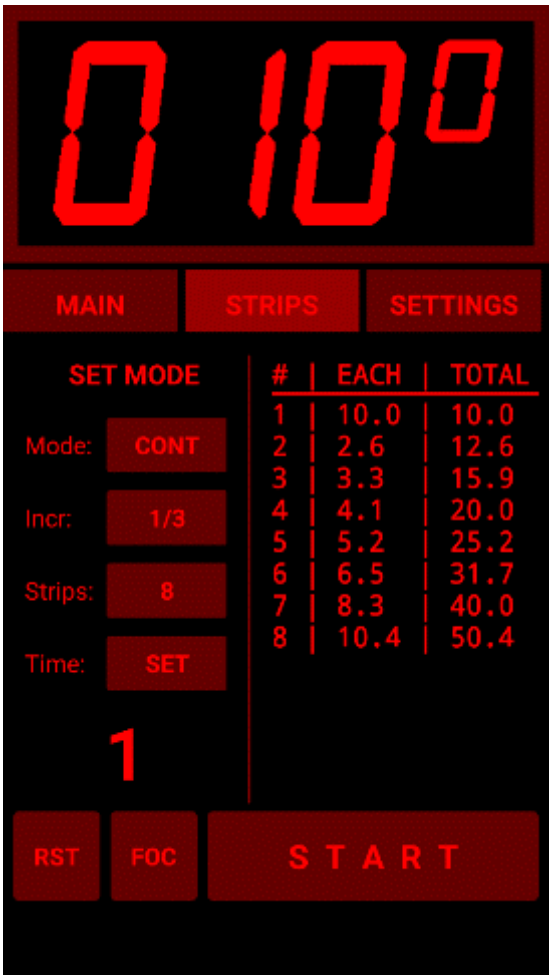
To expose the paper, we use the method where we first expose the whole sheet for the first strip time (= the base time) and then subsequently cover more of the paper with a masking card for each subsequent strip's time.

1) Expose whole sheet for the base time.
2) Move the mask to cover the first strip amount and expose the remainder of the paper for the second strip's time.
3) Move the mask to the next strip amount and expose the remainder of the paper for the next strip's time.
4) Repeat (3) until all the paper is covered.

Tip: *Once you are happy with the configuration, it is worth performing a dummy-run with no paper to ensure you are familiar with what you need to do once the lights are out and there is paper being exposed – especially when there are short strip times involved and you are using CONT mode.*

Examples

1 - [CONT] mode:



If we do not have a Wi-Fi connected lamp switch, we will use this mode and manually switch the enlarger lamp.

On the Settings tab we set a Delay of 3 seconds, so we have time to get ready after starting the timer. We set Mode to CONT (continuous) as we do not have the facility to switch the lamp via Wi-Fi in this example.

We decide that we want the strip increments to be 1/3rd of a stop, we want 8 strips and the first strip is to be for 10 seconds. We select these options and put 10s on the display.

The list on the right of the tab is updated and we note that the times for some of the strips after the initial base time strip #1 are quite short, so we know we will have to pay attention as the beeps will be coming quickly.

When you are ready to expose your paper for the first test strip, the process will be:

1) Tap the [START] button and the timer will tick down the three seconds delay. Once the delay has ended and the timing starts, switch on the lamp - see the tips section below for an idea to use a separate piece of card (or your spare hand) under the lens to perform the timing accurately.

2) Half a second before the end of the first strip's countdown time there will be a short beep followed at the end of the time (0.5s later) by a higher pitched beep indicating the end of this strip and the start of the next. The first beep is a warning to be ready to move the masking card to the next strip position.

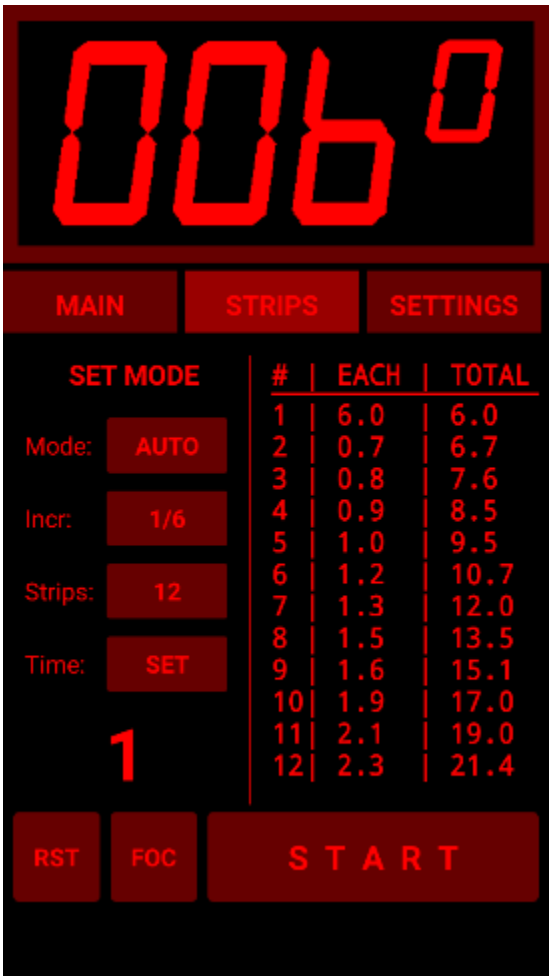
3) At the second beep, move the mask swiftly to the next strip location. The timer will continue timing the next strip with no delay between strips, so you need to move quite quickly.

4) Repeat (3) until all the paper has been covered by the mask.

5) Switch off the lamp.

Process the paper as usual.

2 - [AUTO] mode



This mode requires one of the Wi-Fi connected devices to switch the enlarger lamp on and off. The ideal option if we want multiple test strips on a single sheet of paper.

On the Settings tab we set a Delay of 2 seconds. This time will be used both as a count-in delay as well as the time to pause between strips. We set the Mode to [AUTO] (Automatic) as we can power the lamp on and off and want multiple test strips on a single sheet of paper.

We select that we want the f-stop Increment to be 1/6th of a stop, we want 12 strips and we put 6s on the display for our base time.

The list on the right of the tab is updated and we note that the times for the strips after the initial base time are mostly very short but we do not care as we are switching the enlarger lamp via a Wi-Fi switch and we have set 2 seconds delay to give us plenty of time to move the mask to expose for the next strip time.

When you are ready to expose your paper for the first test strip, the process will be:

1) Tap the [START] button and the timer will tick down the two seconds Delay. Once the delay has ended and the timing starts the timer will switch on the lamp.

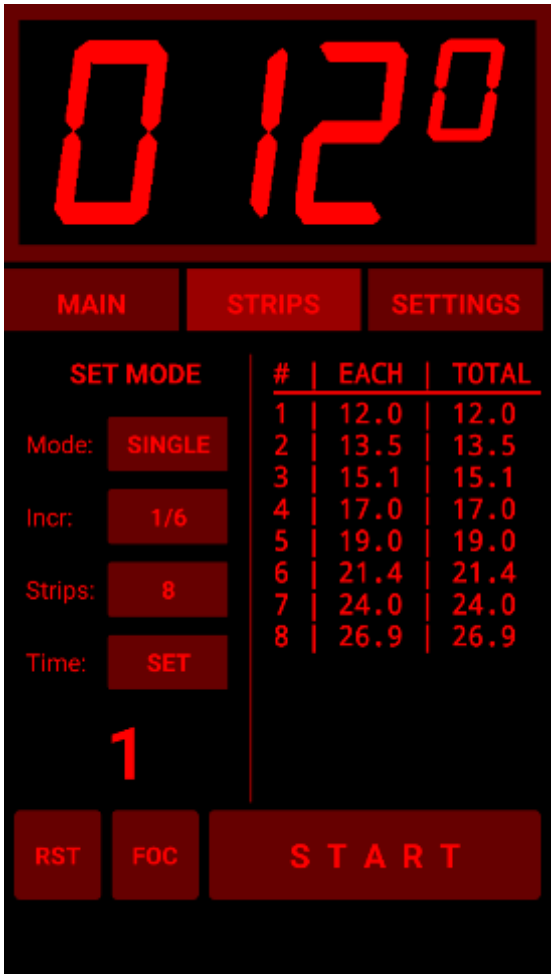
2) When the timer has finished the 1st strip, the lamp will go out and you will have 2 seconds to move the masking card to the next strip position before the timer starts timing the second strip.

3) Likewise, once the current strip has finished, the lamp will go out and the pause time will give you another 2 seconds to move the mask to the next strip position before the timer starts the next strip.

4) Repeat (3) until all the paper has been covered by the mask.

Process the paper as usual.

3 – [SINGLE] single mode



Generally used to expose the same section of a negative for fine-tuning, or if there is too much variation in shadow and highlight detail across the scene to allow us to obtain a good range of tones by having the test strips across the negative. To expose the test strips, we would use either several individual strips of paper, or use a special mask that allows us to expose multiple strips from the same part of the negative on a single sheet of paper.

On the Settings tab we set a Delay of 0 seconds if we are using Wi-Fi connected switching, or a short delay if we are switching the enlarger lamp manually.

We set the Mode to [SINGLE]. We select the f-stop Increment to be 1/6 of a stop, we want eight strips and we put 12s on the display for our base time.

The list on the right of the tab is updated and we note that the strips will range from 12 seconds to 26.9 seconds. We further note that the EACH column is the same as the TOTAL column because in this mode, each strip is for the total f-stop adjusted time.

When you are ready to expose your paper for the first test strip, the process will be (assuming we are attached to a Wi-Fi switch):

1) Tap the [START] button and the timer the timer will switch the lamp on for the duration of the base time (12s in this example).

2) When the timer has finished the 1st strip, the lamp will go out and will wait for you to tap the [START] button again before it commences the 2nd strip.

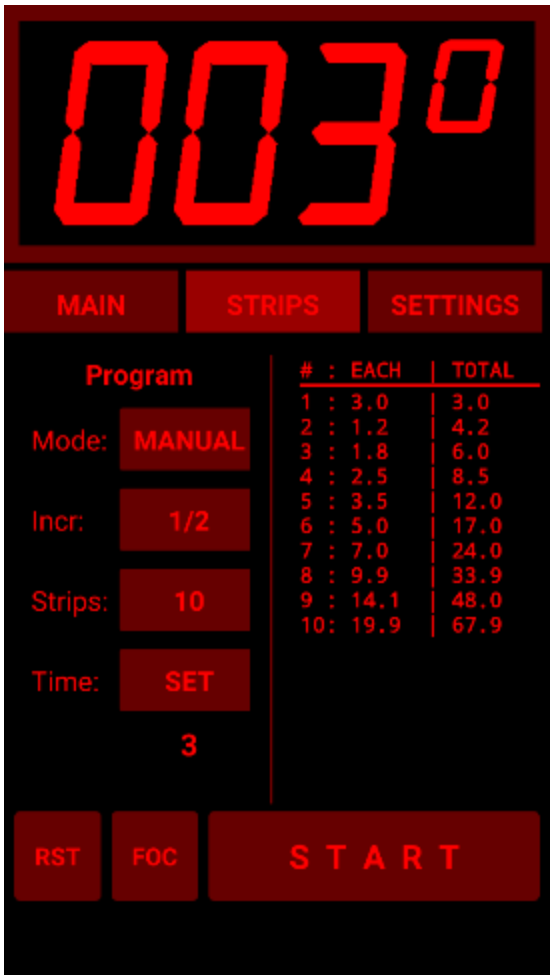
3) Depending on how you are using this mode, you will either replace the test strip with an unexposed strip of paper or will move the paper to cover the same section of the negative with the rest of the paper covered by a window mask. The timer will have advanced to the next strip time, so tap the START button again.

4) Repeat until all the strips have been exposed.

If you are not using a Wi-Fi connected switch, use the Delay option in the Settings tab to give you a short time to prepare to manually switch the enlarger lamp as for other modes other than Auto.

Process the paper as usual.

4 - [MANUAL] manual mode



This mode can be used if you are not sure how long a gap you will need between strips. The timer will stop between strips and wait for you to tap the [START] button. If you are not using a Wi-Fi connected switch, use the Delay option in the Settings tab to give you a short time to prepare to manually switch the enlarger.

On the Settings tab we set a Delay of 0 seconds as we do not require any count-in delay for this example as we will assume the Wi-Fi switch is connected.

We set the Mode to [MANUAL]. We select the f-stop Increment to be 1/2 of a stop, we want 10 strips and we put 3s on the display for our base time.

The list on the right of the tab is updated and we note that the strips will range from 3 seconds to 67.9 seconds. The first few strips after the initial base time are quite short so we may have trouble manually switching so we may want to close the lens a couple of stops and use a longer base-time if we are manually switching, but if we are using Wi-Fi lamp switching hardware, then we will be fine.

When you are ready to expose your paper for the first test strip, the process will be:

1) Tap the [START] button and the timer the timer will switch the lamp on for the duration of the base time (3s in this example).

2) When the timer has finished the 1st strip, the lamp will go out will wait for you to tap the [START] button again before it starts to time the 2nd strip.

3) Likewise, once the current strip has finished, the lamp will go out and the timer will wait for you to tap the [START] button again before it starts to time the 2nd strip.

4) Repeat (3) until all the paper has been covered by the mask.

Process the paper as usual.

Tips:

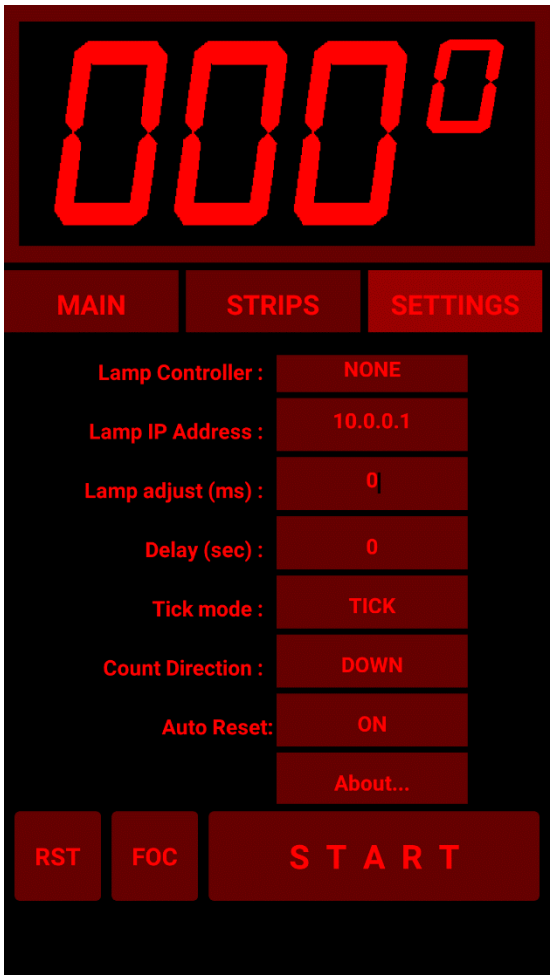
i) Putting a series of short lines on the paper at the strip positions using a marker pen can help you locate each strip position quickly when exposing the paper. This is also useful if using small f-stop increments where the differences between strips may be very small, making it difficult to identify where one strip ends and the next one starts – especially in busy areas.

ii) When switching the lamp manually, there is a slightly more accurate (and less panic inducing) way than using an external switch to power the lamp directly: use a piece of black card under the lens. That way, you can switch the lamp on while holding the card under the lens and then move it out of the way quickly when the timing starts and move it back again when the timing ends before switching off the lamp. Setting a two or three second Delay helps here. This may seem like an extra bit of faff, but it is less fraught and removes the threat of vibration as you press a switch – plus you will already have a suitable card in your hand to use for masking each strip. I used such a method with a metronome as my only timing method for years - using the f-stop timer app is a bit of a jump from that but the black card is still useful (even though the metronome is now confined to the back of a cupboard).

iii) If you have no idea what exposure you will need, set the F-stop adjust to ½ stop (or even 1 stop) and set the base-time to 3 seconds and use a quarter-sheet of paper with 6-8 strips to get within ½ stop of the required exposure and then use this information to perform a second test strip with smaller f-stop increments to find the precise time. Using a base time of 3 seconds and ½ stop increments, with 8 strips you can span 3s to 48s which should get you in the ball-park for a more precise test.

iv) In practice, once you get used to this method, you don't really need to set the number of strips. Setting 20 Strips all the time is OK as you then just step the masking card across the paper for each strip until you run out of paper. As long as you can clearly see where each strip starts and ends (not always possible with small f-stop increments) you can refer to the list on the timer's display for the total time taken for each strip.

Settings Tab



This tab is scrollable, so if you do not see all the options, scroll down and they will appear.

Lamp Controller: Selects between the various hardware switching controller options (and NONE).

If you are not using any lamp switching Wi-Fi hardware select [NONE]. [None] is selected by default when the timer is first installed, but any change you make to this value is stored in your device's memory thereafter and reloaded each time you restart the timer app. Selecting [LCTECH] will (from version 1.5.0) pop up a warning dialog to avoid unexpected error messages from Android.

Lamp IP Address: When used with a hardware interface, this is the IP address of the Wi-Fi connected lamp switching hardware.

This address defaults to 10.0.0.1 but can be changed and the change will be stored for future use. If the Sonoff device 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 device using port 8080 on IP address 10.0.0.4).

Tip: I find it best to connect the phone or tablet to the Sonoff or LCTech Wi-Fi network access points before starting the timer app to ensure the timer app connects to the hardware correctly, otherwise you may need to restart the app to make the connection.

Lamp Adjust: When a lamp switches on it takes some time to reach full brightness. When the lamp then switches off, it takes some time to decay to full darkness. These times, and the precise ramping up and down of the brightness usually differ and hence the amount of light that reaches the photographic paper is rarely the same when the lamp reaches maximum brightness for ON and dims down to zero for OFF actions. There are also potential delays introduced by voltage stabilising and other lamp driver circuitry when in use - LED lamps are a special problem in this area.

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 overall exposure. For example, if we have a difference equivalent to 1/3rd of a second less exposure, four individual 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 equivalent to 10.67 seconds - short by 1/6th of a stop. A 1/6th 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 exposure time 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 exposure time and a negative value subtracts that number of milliseconds.

This value is automatically saved to the Android device whenever it is changed and 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 with a one second pause between exposures (to mimic the STRIPS tab 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 if 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 once the paper has dried. If the 10x2-second exposures sheet is lighter than the single 20-second exposure sheet then you need to make a +ve adjustment and if darker, you will need to make a -ve adjustment. It may take several iterations to find a precise value.

You can avoid all this by simply letting the timer step-through the individual test strip times used to reach the required time once the correct time is found. For example, if you find the correct time is on strip 5 using the AUTO mode, set the number of strips to 5 and start the timer and let it run through each strip time in turn for the whole sheet (or change to MANUAL and step through them one by one).

Delay: You can set a 0 to 10 second delay before the timer starts its countdown after hitting the [START] button. This defaults to 0 seconds and is retained in the device's memory if changed. The timer will tick the seconds. This allows you to prepare before the timer starts. Useful if you are switching the lamp manually to give you time to prepare. Also used in the Strips tab to provide a delay between strips in AUTO mode.

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

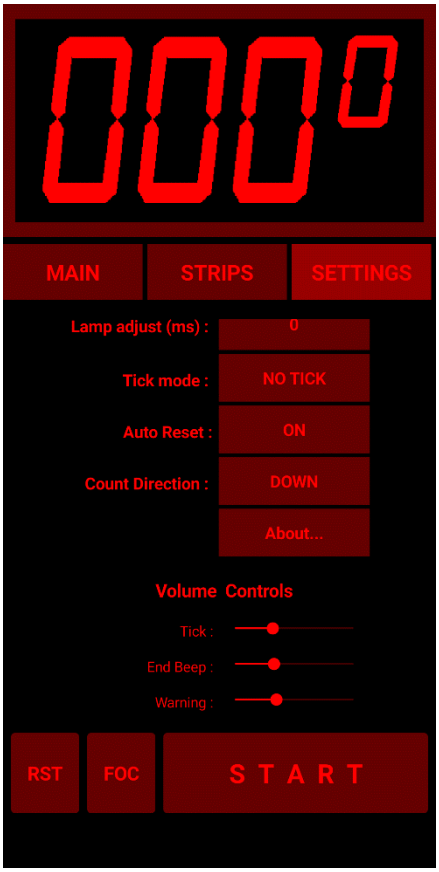
> NO TICK: timer does not tick.
> TICK: The timer ticks every second.
> LAST 5 SECS: the timer will tick off the last 5 seconds of the countdown but only if the time was for 10 seconds or more. Can be useful to alert you to the fact that the timer has nearly finished if you are switching the lamp manually.

Count direction: Rarely useful - 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 and none of the timer's functions work in count = UP mode. The timer app always starts in count direction = DOWN.

Auto Reset: Provides the option to automatically reset the timer display to the previous countdown time value at the end of the current countdown. A short tap of the [RST] button will still work to perform the same operation if this is set to OFF. In practice auto-reset ON is by far the most useful option and the option to set it OFF may be removed in future.

About: opens a screen with release notes, topical information and links to the manual and code on GitHub.

Volume Controls: Individual controls for Ticks, the End Beeps and the [CONT] strips mode warning beep. A bit tricky to use due to their small size - I find it helps to touch the small button and wait a second before sliding it. Provides audio feedback. Values are stored and reloaded on restart.



Appendix 1: Connecting the App to the Lamp Switching Hardware

If using one of the hardware Wi-Fi connected lamp switching options, you will need to configure the app to connect to the device by selecting the appropriate hardware interface.

Sonoff Basic R3 and Mini In DIY Mode

This mode does not require reprogramming of the Sonoff devices as these have a DIY mode that allows them to be switched via commands sent within your local network. The Sonoff S20 and S26 smart sockets do not have DIY mode so cannot be used in this way. There are two timer modes that use the DIY mode detailed below.

First, place your R3 device into DIY mode per Sonoff's instructions This process is documented by Sonoff here:
http://developers.sonoff.tech/sonoff-diy-mode-api-protocol.html#eWeLink-Mode-and-DIY-Mode
but below is a more detailed look at the specific process we need to follow.

Enter the Compatible Pairing Mode (AP) by pressing and holding the button down for 5 seconds until the LED changes to continuous rapid flashing.

Looking at your available networks, you should see a new network called ITEAD-XXXXXXXXXX appear (where the X numbers will depend on your specific device) after a few seconds. Connect to the ITEAD-XXXXXXXXXX network using the default password 12345678

Once connected, use a browser to browse to address 10.10.7.1 and a page with a button should open asking you to set your network login. Click the button.

> *Tip: If this does not appear, try entering the device request as follows:* 10.10.7.1/device *in the browser and if you get a response, you probably need to reset the Sonoff and upgrade its factory firmware using eWeLink and start again. Restart the Sonoff, pair to eWeLink, perform any update to at least v3.5 and then Delete it from eWeLink - this will force a factory reset. If you do not get a response from the* 10.10.7.1/device *request then something else has gone wrong so check that you still have the rapidly flashing LED signal and have not lost the connection to the device.*

Enter your network SSID name and password as requested and click on "Save". When the device restarts it should be found on this network.

The browser should now display "SUCCESS" - but - it is only saying that it found your network, it may still not succeed in logging in to it.

> *Tip: When the device has connected to your network the LED will change to repeating two short flashes. If it does not log in OK, either you entered something incorrectly or it may be trying to connect to a 5GHz network rather than a 2.4GHz one. Reset the device and try again.*

Once it has connected to your network, you can relax as there is no longer any need to rush - it will stay connected until you switch it off and will reconnect the next time you power it on.

Find the new device's IP address. I suggest connecting your phone or tablet to the same network and using the Fing app.

> *Tip: On my network, the new device does not appear in my router's list of connected devices despite the fact that it got its IP address from the router's DNS service. I do not know if this is a "feature" of my router or a consequence of the mDNS protocol used by the Sonoff in DIY mode. In any event, connecting my phone to the same network and using the Fing app detected the Sonoff device. To confirm that you have found the correct device, select the device and use the "Find Open Ports" option and Fing should find a "blackice-icecap" user console on port 8081.*

Once configured and working in DIY mode, make sure your Android device is connected to the same network and set the IP Address of the Sonoff device in the IP ADDRESS section in the SETTINGS tab (just the IP address, do not add the port etc as this is added in the app's code): e.g. **192.168.0.22**

> *Tip: I had to use the Fing app on my phone to find the device's IP address - it did not appear on my router's list of connected devices for some reason.*

> *Tip: if you make a mistake setting DIY mode, or if your R3 device has previously been connected using the eWeLink app, you may find you cannot set the device into DIY mode. If this occurs, connect the device to the eWeLink app again and then Delete it from the eWeLink. This seemingly pointless activity will force a factory reset of the device and allow you to set it in to DIY mode as if it was a new out-of-the-box device.*

> *Tip: another reason for DIY mode not working is if you have an older version of the Sonoff firmware. Use the eWeLink app to check and, if necessary, update the firmware to at least V3.5. Remember to Delete it from the app after updating*

.

PULSE R3 mode:

In this mode the timer sends commands to the Sonoff device to switch the lamp on for a given amount of time using the *Inching* API call (http://developers.sonoff.tech/basicr3-rfr3-mini-http-api.html#Inching). This means the timer does not need to send the OFF signal. Not sending the OFF signal means timings are not affected by random delays caused by the network router being too busy processing other devices on the network when we send ON and OFF commands separately.

Unfortunately, and annoyingly, the Sonoff API only allow multiples of half a second to be set for the timing in this mode. This hurts accuracy somewhat but is not as bad as it first appears for typical lamp ON times of several seconds and it does at least mean that the timing is consistent which you could argue is more important.

The worst-case error when rounding to the nearest 0.5s is 0.2s (10.7s is rounded to 10.5s, 10.8s is rounded to 11.0s, etc). The final percentage error depends on your base time: for example, 10.7 seconds will round down to 10.5s which is an error of only 1/37th of a stop which is not detectable by the human eye. Longer times will be more accurate but shorter times will be worse: 5.3s for

example would round to 5.5s which is an error of 1/19th of a stop - which I think is still unlikely to be perceptible. Timings of under 0.3 seconds will be rounded to 0.0 seconds so the enlarger lamp will not light for this strip - the timer will beep at the end of the strip delay time however so you will still know to move the masking card to the next strip.

I prefer the repeatability of the timings in this mode over the random router dependent delays introduced using [BASIC R3] mode below and I think this is currently the best option for people who do not wish to reprogram their Sonoff Basic R3 or Mini.

BASIC R3 mode:

The downside of this mode is that all the lamp's ON and OFF signals go via your Wi-Fi router separately which adds random delays of typically 5 - 100ms on my network, and occasionally much more.

In my testing on my network, this mode was not useable as approximately once in every ten times a delay of over 1/4 of a stop was randomly generated. I used a microphone placed next to the Sonoff device and Audacity running on my PC to view the clicks. Sending a series of Strip times to the Sonoff and measuring the time between ON and OFF clicks showed good accuracy most of the time but occasionally a random unacceptable time would appear.

If you have a spare Wi-Fi router then creating a separate network and dedicating this to the timer with no other devices or Internet connection would make these delays both smaller and much less random in duration allowing you to use the timer's Lamp Adjust setting to compensate if necessary. I have not tried this, but it should work OK in theory, but I prefer the PULSE R3 mode above.

Reprogrammed Sonoff Basic, Mini, S20 or S26 Wi-Fi Switches

See Appendix 3 or 4 as appropriate for the device model for programming details. You will need to disconnect from your home network and connect to the separate network used by the Sonoff switch. The Wi-Fi 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 suggested that you change this to something more relevant to yourself.

Configuring the Sonoff device as an Access Point (AP) with its own Wi-Fi network and embedded web server in this way has the primary advantage that there are no delays introduced by going through your local network router or (much worse) the Internet. You also do not need to hunt for whatever IP Address your Wi-Fi router's DNS service 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 SSID which is why you should change the password - just in case you have smarter than average neighbours...

To summarise:

1. Connect the reprogrammed Sonoff device to a power outlet and connect your enlarger to 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. Tap the Focus [FOC] button and the enlarger lamp should switch on.

6. Tapping [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 Wi-Fi and connect to the FADUTimer network again after switching to Aeroplane mode. If it still becomes sluggish, restarting the phone will clear out the device’s memory and improve things further. It is always worth doing this before starting a printing session. I have tried the app with 6 different Android devices of various Android versions and vintages (4 phones and 2 tablets) and only one had this issue (a Motorola G5).

LC TECH relay board

In my opinion, you should not connect this board to mains voltages. I do not believe the PCB layout makes this board safe to use at high voltages. It has a track running with only 1mm separation between the relay switching voltage pin (5 volts) and the relay terminal which will be at full mains voltage. Use it only on the low-voltage side of a transformer/voltage stabiliser. Switching the low-voltage side in this way is the preferred option as it can result in less strain on the voltage stabilizer circuitry.

This board does not require reprogramming as it can connect directly to your Android device's Wi-Fi via its built-in Wi-Fi Access Point (AP).

The board has two relays. We are only using Relay-1 here. Relay 2 is reserved for possible future use. There is a single-relay version of this board but it uses different command strings so will not work unless you edit the FADU timer app's code in App Inventor.

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 Wi-Fi Network is called "AI-THINKER_xxxxxx" where "xxxxxx" is a unique 6-digit hexadecimal number (this is the device's MAC address - and it 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 running the timer app or you will get a pop-up error message. To avoid confusion the app now asks you if you are sure you want to connect to the board. 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. Tap the Focus [FOC] button and the enlarger lamp should switch on
5. Tapping [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 Wi-Fi and connect to the AI-Thinker network again after switching to Aeroplane mode. If it still becomes sluggish, restarting the phone will clear out the device's memory and improve things further. It is always worth doing this before starting a printing session.

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. Technically, the term ***f-stop*** only applies to lens aperture, but the end result is the same when you adjust exposure time using the same exponential ratios, and ***f-stop printing*** has become the usual term.

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 same 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.

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: Hardware Programming Older Versions of the Sonoff Basic and all S20/S26 Devices.

Newer versions of the Sonoff Basic and Mini (Basic R3 and Mini versions) can be Over The Air (OTA) programmed without need for extra programming hardware. This is detailed below in Appendix 4 but older versions and all versions of the S20 and S26 sockets do not have OTA options. Also, if you lose your R3 version's factory firmware this will be the only way to get the new timer firmware on those devices.

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.

The Sonoff Basic and S20/S26 devices are reprogrammed in exactly the same way – the only difference being the connections for the FTDI module on the Sonoff PCB.

Note that once programmed, the original firmware for the Sonoff 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 Basic, S20 or S26 Wi-Fi enabled smart socket.
3. FTDI adaptor: 3.3v version.
4. USB cable for the FTDI board: make sure it can sync – 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 etc. Make sure you get the correct S20/S26 smart socket for your country's power socket type. One reason for choosing the Sonoff S20/S26 is because they are available in several regional variations – the above shows the UK type.

The FTDI adaptor (so-named after the manufacturer of the original and most popular chip used in these boards: "FTDI" https://www.ftdichip.com) is more accurately called a "USB to serial port adaptor". The FTDI adaptor used for this project MUST be 3.3V compatible. Many available adaptors 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 Sonoff device will kill it.

Prepare the Sonoff S20

The same instructions are suitable for the S26 - the only main difference is a small change in the connector pinout on the Sonoff S26 board.

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 carefully 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 board on the ones I have had is very tightly fitted and needs *GENTLY* levering up next to the screw holes after removing the screws – use something plastic as a spudger. 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 as it easily pings out (you can probably guess how I know that!).



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

O	Gnd
O	Tx
O	Rx
O <	Vcc

Prepare the Sonoff Basic

The Sonoff Basic easily comes apart by unscrewing the four screws holding the end-plates and pushing the base of the case out. This then exposes the strip of five pcb holes to solder the header pins to. The key pin ">" is the one closest to the switch with the long actuator shaft to the left of this picture.
Note that the pins are in a different order to the S20 (and we only use the top four pins). The image below shows the location of the socket after being soldered in:



> O	Vcc
O	Rx
O	Tx
O	Gnd
O	not connected

A more recent version of the Sonoff Basic uses a 4-pin connector as shown on the right, but it uses the same pin order as above.

[O]	Vcc
O	Rx
O	Tx
O	Gnd



In both cases (Sonoff Basic and S20) these need to be connected to the corresponding pins on the FTDI adaptor using the Dupont connector wires:

Sonoff to FTDI Pins:

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

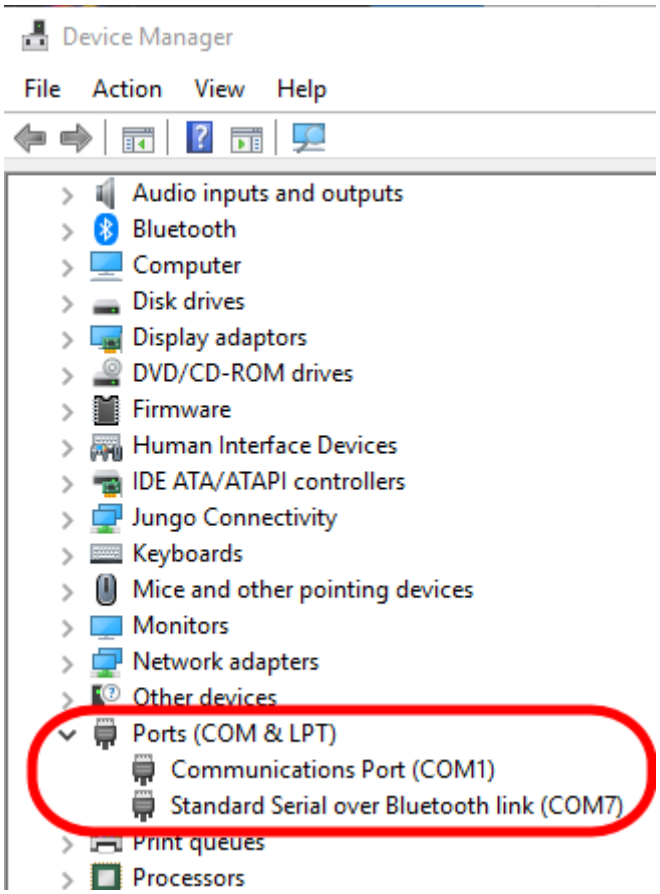
Other pins on the FTDI adaptor are left unconnected.

Tip: if you use Windows, get an FTDI adaptor that uses an actual FTDI chip. As previously mentioned, "FTDI" is 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 extra faff to step through to find and load the drivers.

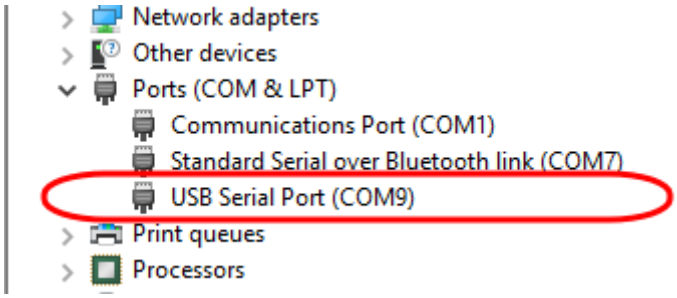
Prepare the FTDI adaptor

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 via USB while pretending to be a serial port to your computer).

I am not familiar with Macs and Linux based PCs, but there is a lot of info on this online. 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 an 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:

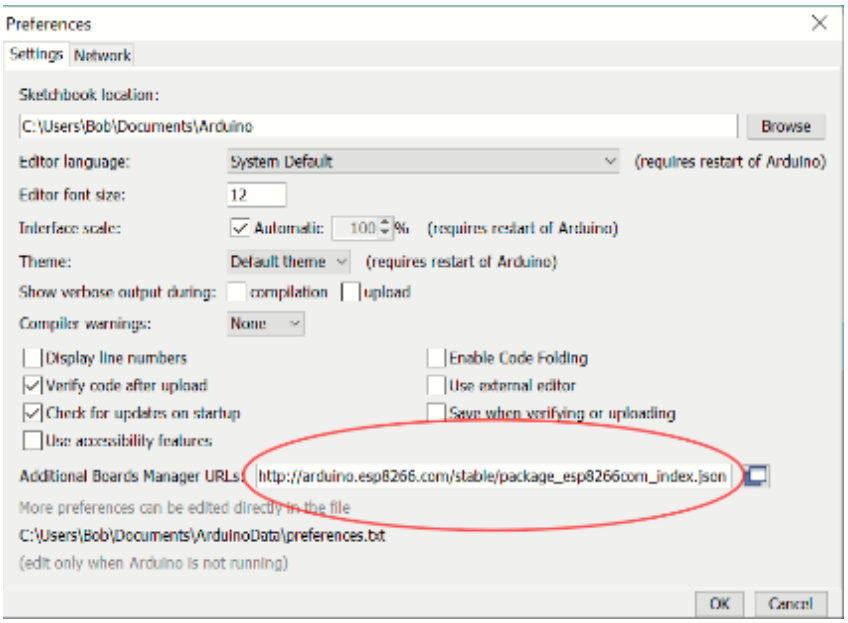


When you can see this extra Com port, your FTDI board has been recognised and is ready to use. If after refreshing the Device Manager it is still not recognised you will need to troubleshoot, in which case google.com is your friend – just search for: FTDI Arduino problem.

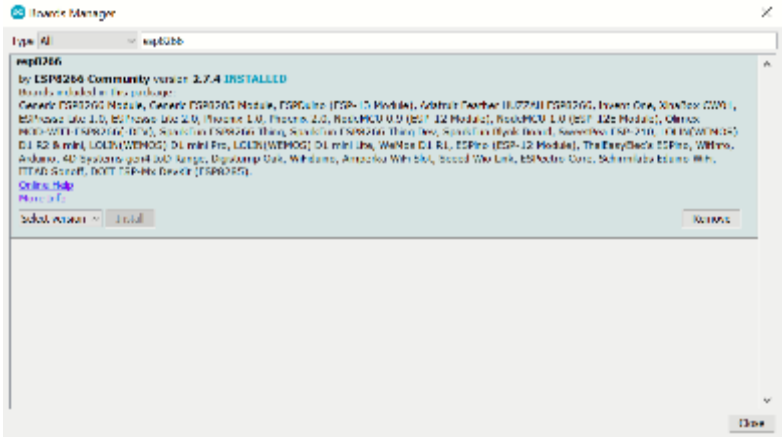
Prepare the Arduino IDE

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

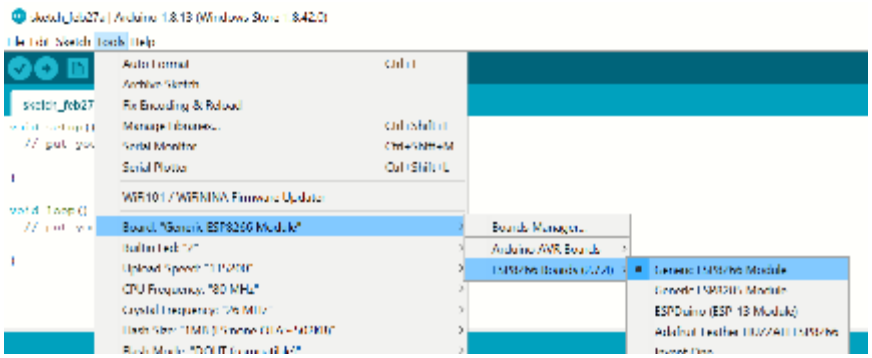
Run the Arduino software and from the menu bar select the ***File*** menu and then ***Preferences***. In the ***Additional Boards Manager URLs:*** box enter:
http://arduino.esp8266.com/stable/package_esp8266com_index.json and click ***OK***.



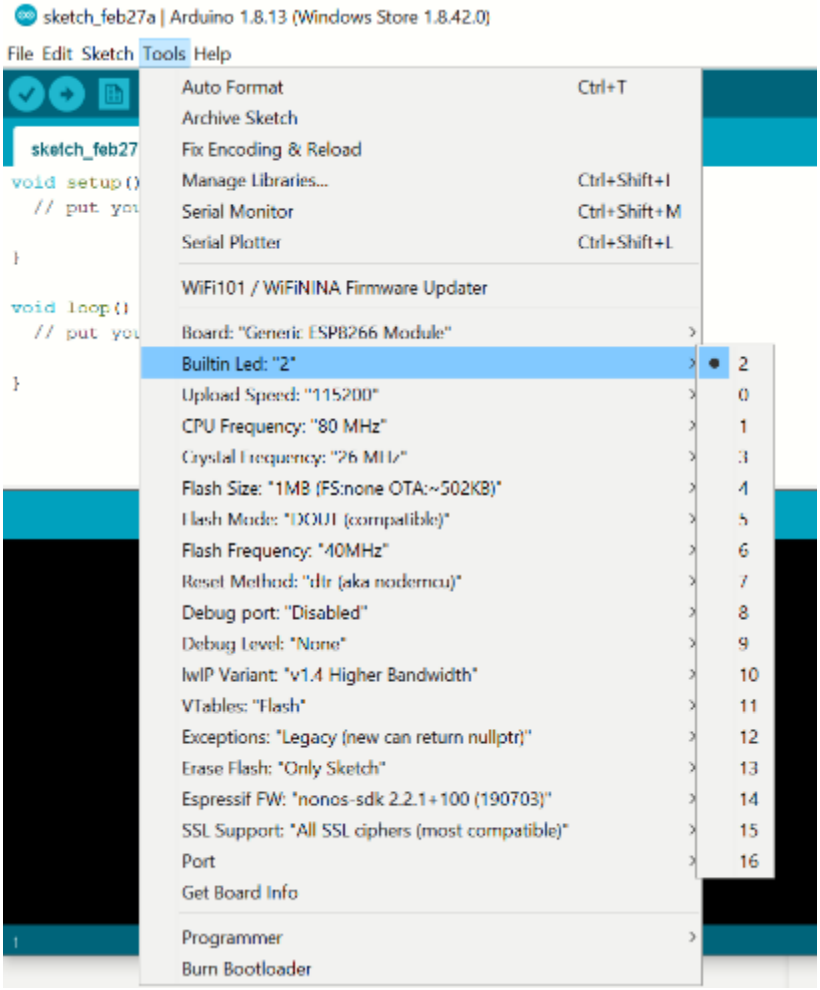
Now we have told the Board Manager where to look for new board configurations, open the Boards Manager itself (Tools menu => Boards Manager). In the window that pops up, scroll past any other items and select the ***ESP8266 by ESP8266 Community*** item. Click Install and several ESP based board configurations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



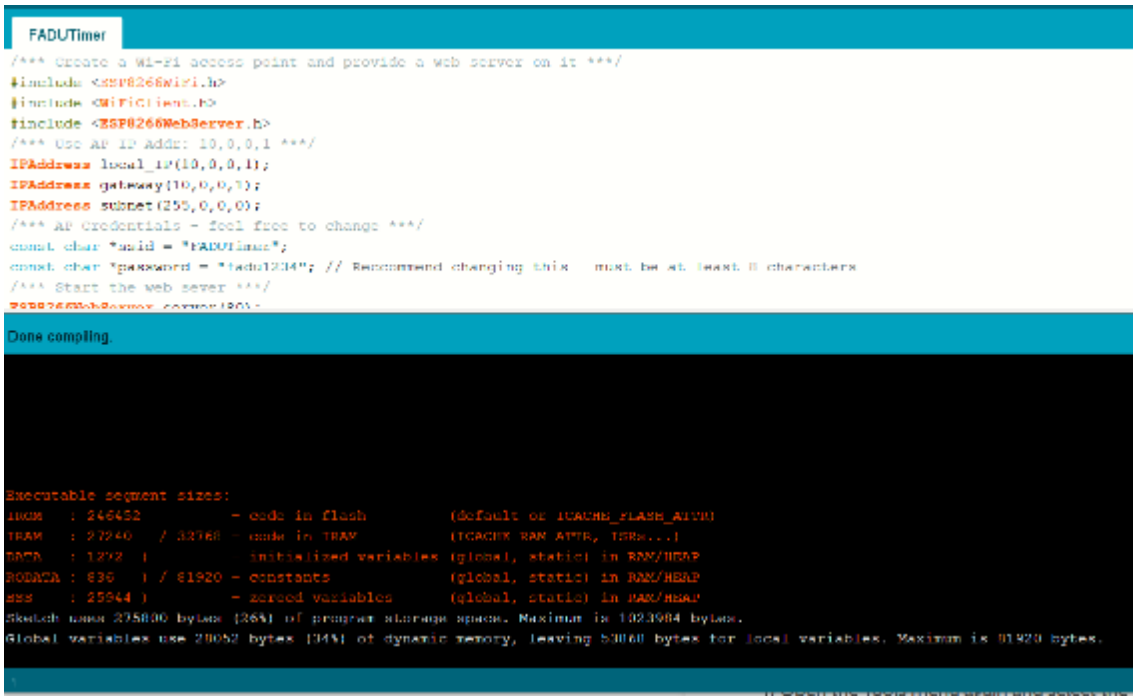
Also note that some of the other settings on this menu may need changing. You should set it to the settings shown below (except the ***Port*** setting which will need setting to the port your FTDI adaptor has been mapped to):



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 Sonoff device, 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 ESP82666 chip's resources used by the

program and no error messages. Pasting text into the Arduino software's editor often introduces additional and unexpected new lines in the middle of text which can throw errors so check for that if you get errors - clicking on the error warning will usually indicate the location of the error in the editor window above.



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

Programming the Sonoff Basic or S20 devices

1. **MAKE SURE THE SONOFF DEVICE IS NOT PLUGGED IN TO THE MAINS**. The FTDI adaptor will supply the power to program the device for programming and if the device is plugged into the mains, you may not only blow the FTDI board but will very possibly fry your PC and/or yourself too if you touch either.
2. If you are soldering the header pins to the device, do so now.
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 Sonoff 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 Sonoff is dead. 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 on the Sonoff device and, keeping the switch pressed, plug the FTDI board into your computer. This puts the ESP8266 Wi-Fi chip in the Sonoff device into programming mode. After a second or two release the switch, leaving the FTDI adaptor connected. The board is now ready for programming.
9. In the Arduino IDE, open the sketch menu and If not already compiled, hit the right-pointing arrow below the menu bar to first compile and then upload the new firmware. Compiling and uploading the firmware will take approximately a minute.
10. Once the new firmware has been uploaded, unplug the FTDI module from your computer and remove the cable from the Sonoff's header and reassemble the device. Before unplugging it, you may see the FADUTimer network appear on your phone's list of available networks, but do not try to connect to it as your FTDI adaptor probably cannot supply enough power to drive the relay.
11. Connect the Sonoff to the mains, connect to its Access Point (FADUTimer by default) and use the timer app's [FOC] button to see if you can hear the device's relay switching.

The new firmware:

```
/*** Create a Wi-Fi 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 = "fadu1234"; // 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 lamp control commands ***/

/* "/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 - is an example of re-flashing the Sonoff Basic device.

Appendix 4: OTA Programming Basic R3 and Mini

Using the OTA (Over The Air) Process to Re-program the Sonoff Basic or Mini R3.

Currently (February 2021) only the Sonoff Basic R3 and Mini R3 are suitable for this - the S20/S26 smart socket devices do not have the necessary OTA mode.

Instructions on using OTA to program the Sonoffs was kindly provided by Ariel Till <tillari@yahoo.com> who provided instructions on how to update the firmware using OTA. His original instructions in Spanish are here: https://www.lasnuevemusas.com/timer-...elefono-movil/.

I have provided the instructions in English below based mainly on Ariel's instructions and my own experience when following them. I have added some notes and images to help clarify for those with no experience with these technologies.

Note that the Sonoff can only connect to a 2.4GHz network. Most modern Wi-Fi routers will provide both 2.4 and 5GHz networks using the same SSID name and bridge between them. If your networks have different SSIDs (unusual, but someone may have made the change) you must connect the Sonoff device to the 2.4GHz network SSID.

If your OTA update fails part way through you may find you have a junk Sonoff with no firmware at all on it. You cannot recover the original Sonoff firmware so make sure you do not interrupt the process once it starts. If using a battery powered device to program the new firmware make sure your battery has plenty of charge. The worst-case scenario is that the only way to use the Sonoff after such a failure will be to use the FTDI hardware method described elsewhere in this manual to flash the new FADU timer firmware - this should still work as it uses the ESP8266 WiFi chip's hardware flashing process.

You will require:

1) A Sonoff Basic R3 or Mini.
2) The Chrome browser with the RESTer plugin installed.
   We need this to talk with the Sonoff API via POST requests. You could use Postman instead but if you are not familiar with this process, it is probably best to stick to the instructions here using RESTer.
3) The nginx web server.
   I think this only runs on the Windows operating system. If you can place the firmware code on another web server that you already have access to, you can do that. nginx is required because the Sonoff device needs to download the new firmware file from a web server. Luckily, nginx is simple to install and does not require any configuration to use for our purposes.

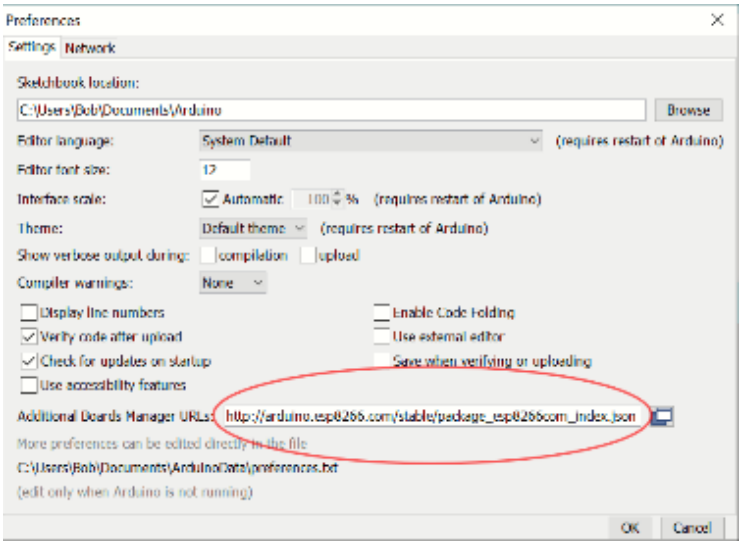
If necessary, install Chrome and add the RESTer extension and the nginx software before you start and make sure the Sonoff device powers up in Pairing mode (two short flashes followed by a long flash, repeated).

You may need to connect to the eWeLink app to update the firmware to version 3.5 or higher. Also, if you have previously paired your Sonoff device, you may need to do a factory reset using the eWeLink app. To factory reset the device, add it to eWeLink in the usual way and then Delete it -

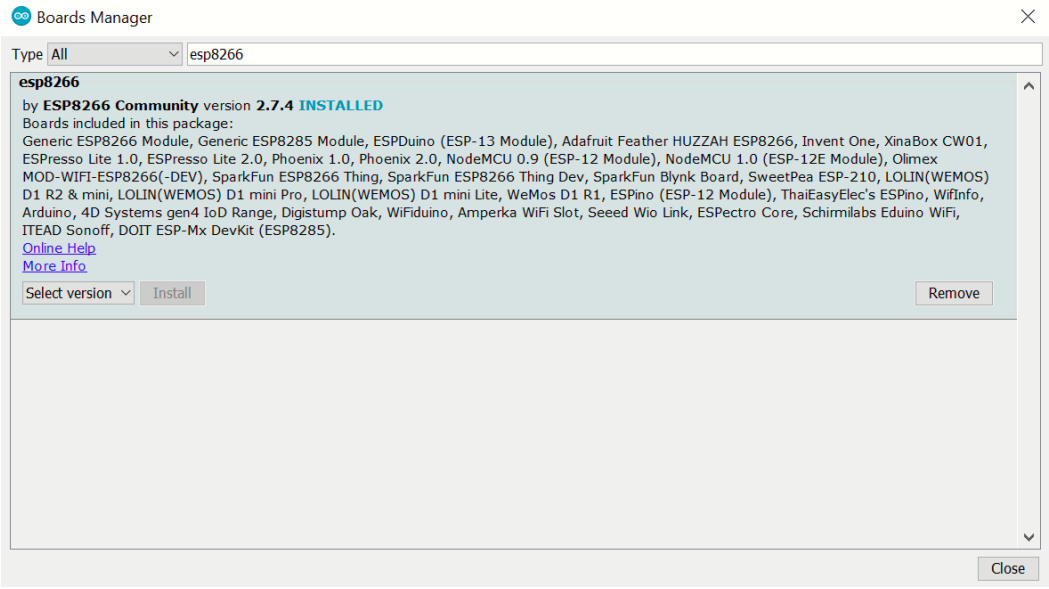
this will force a factory reset. To play safe regardless of whether you are using a brand new or previously paired Sonoff device, I suggest pairing with the eWeLink app to check the version and upgrade it if necessary and then Delete the device from eWeLink and hence also force the reset even if it is a new device. After reset, remove the unit from mains power for a few seconds and then power it back on. It should restart in quick-pairing mode (two quick flashes and one long one) indicating that it is waiting to be paired to.

1) Download and install the latest Arduino IDE software from https://www.arduino.cc/en/Main/Software

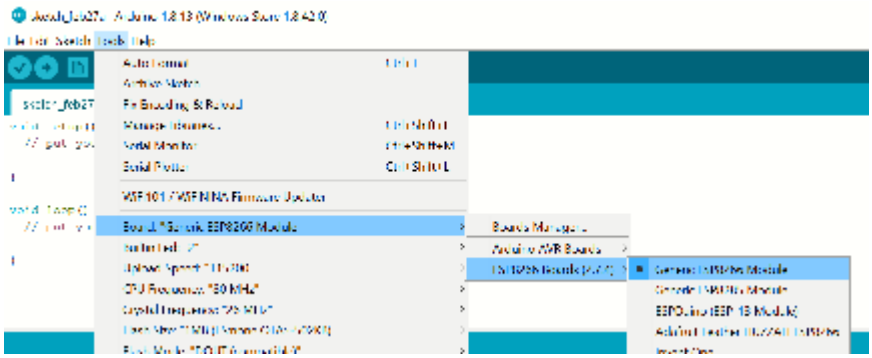
2) We need to add libraries for the ESP8266 WiFi chip used by the Sonoff devices, so run the Arduino software and from the File menu select Preferences. In the window that pops up section: "Additional Boards Manager URLs", enter http://arduino.esp8266.com/stable/package_esp8266com_index.json and click OK.



3) Open the Card Manager from the Tools => Board menu. In the window that opens, find and select the item "ESP8266 by ESP8266 Community", and click Install. Once the install has completed, click on Close.

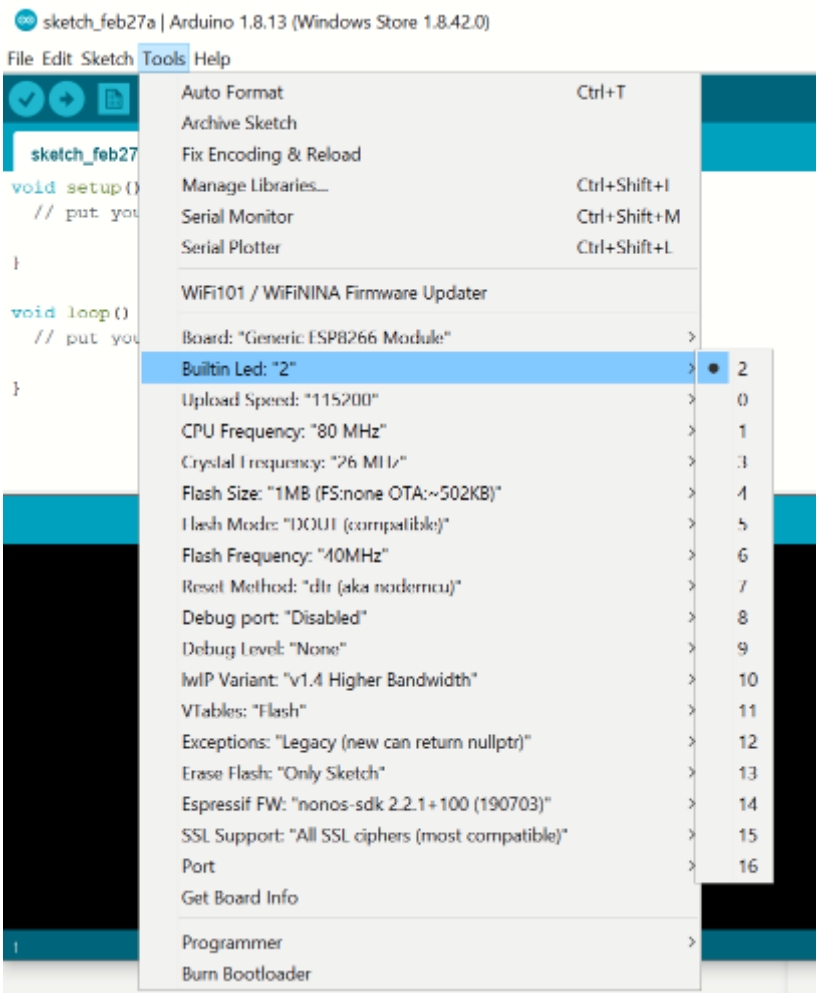


4) Open the Tools menu again and select the Generic ESP8266 Module board.



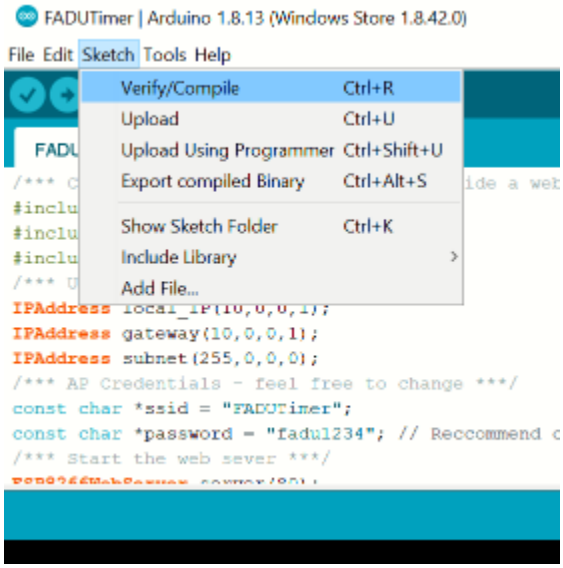
5) Open the Tools menu once more (not for the last time!) and select the following from the new settings that have appeared (the menu closes each time you make a change - annoying, but as expected):

Builtin Led: “2”
Upload Speed: “115200”
CPU Frequency: “80 MHz”
Crystal Frequency: “26 MHz”
Flash Size: “1MB (FS:none OTA:~502KB)”
Flash Mode: “DOUT (compatible)”
Flash Frequency: “40MHz”
Reset Method: “dtr (aka nodemcu)”
Debug port: “Disabled”
Debug Level: “None”
IwIP Variant: “v1.4 Higher Bandwidth”
VTables: “Flash”
Exceptions: “Legacy (new can return nullptr)”
Erase Flash: “Only Sketch”
Espressif FW: “nonos-sdk 2.2.1+100 (190703)”
SSL Support: “All SSL ciphers (most compatible)”

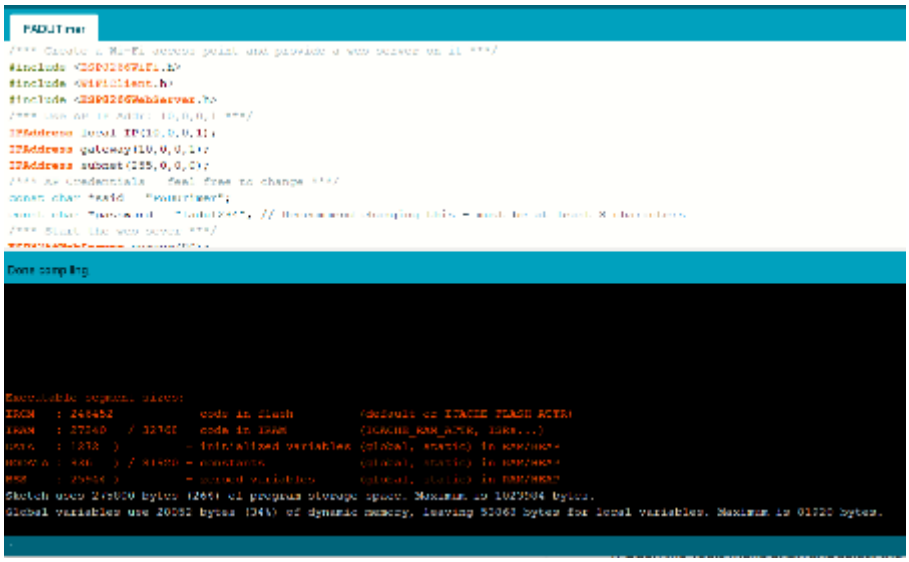


6) In the Arduino IDE code window, delete the default code and paste the new code for the timer provided below. Select the Save As option from the File menu and save the new sketch code.

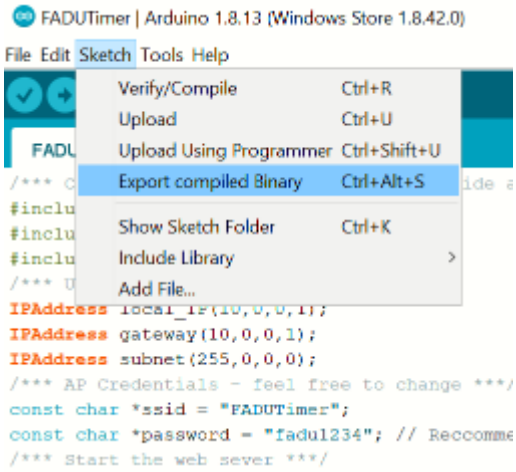
I would suggest testing a compile of the code using the Sketch => Verify/Compile menu option as it is likely an error or two may have crept in when pasting from this document to the Arduino IDE editor - it often inserts unwanted new-lines, Returns, etc.



Make any changes you wish to the password etc and once any compile errors have been fixed, save the code again for any future reuse.



7) Open the Sketch menu again and select the "Export Compiled Binaries" option. This will create a .bin file with the compiled code which can be found in the same folder as the source code you just saved - e.g. if you saved the code as FADUTimer it will be called: FADUTimer.ino.generic.bin



8) We are now ready to prepare to program the Sonoff with the new firmware using the RESTer plug-in running on the Chrome browser.

If not already done, run the Chrome browser and install the RESTer Chrome Extension.

Power cycle the Sonoff (i.e. unplug it and plug it in again) and while the device is in pairing mode (two short flashes followed by a long flash) press and hold the Sonoff button for 5 seconds until the LED changes to flashing rapidly - this shows that it is now in API pairing mode. You have a couple of minutes to link to it before it gets bored of waiting and the rapid flashing changes and you have to power cycle it again and retry.

This process is documented by Sonoff here:
http://developers.sonoff.tech/sonoff-diy-mode-api-protocol.html#eWeLink-Mode-and-DIY-Mode

but the below is a more detailed look at the specific process we need to follow.

Entering the Compatible Pairing Mode (AP) by pressing and holding the button down for 5 seconds until the LED changes to continuous rapid flashing.

Looking at your available networks, you should see a new network called ITEAD-XXXXXXXXXX appear (where the X numbers will depend on your specific device) after a few seconds. Connect your computer or other device that you are using Chrome on to the ITEAD-XXXXXXXXXX network using the default password 12345678

Once connected, use a browser to browse to address 10.10.7.1 and a page with a button should open asking you to set your network login. Click the button.

> *Tip: If this does not appear, try entering the device request as follows:* 10.10.7.1/device *in the browser and if you get a response, you probably need to reset the Sonoff and upgrade its factory firmware as described above using eWeLink and start again. If you do not get a response from the* 10.10.7.1/device *request then something else has gone wrong so check that you still have the rapidly flashing LED signal and have not lost the connection to the device.*

Enter your network SSID name and password as requested and click on "Save". When the device restarts it should be found on this network.

The browser should now display "SUCCESS" - but - it is only saying that it found your network, it may still not succeed in logging in to it.

> *Tip: When the device has connected to your network the LED will change to repeated two short flashes. If it does not log in OK, either you entered something incorrectly or it may be trying to connect to a 5GHz network rather than a 2.4GHz one. Reset the device and try again.*

Once it has connected to your network, you can relax as there is no longer any need to rush to will stay connected until you switch it off and will reconnect the next time you power it on.

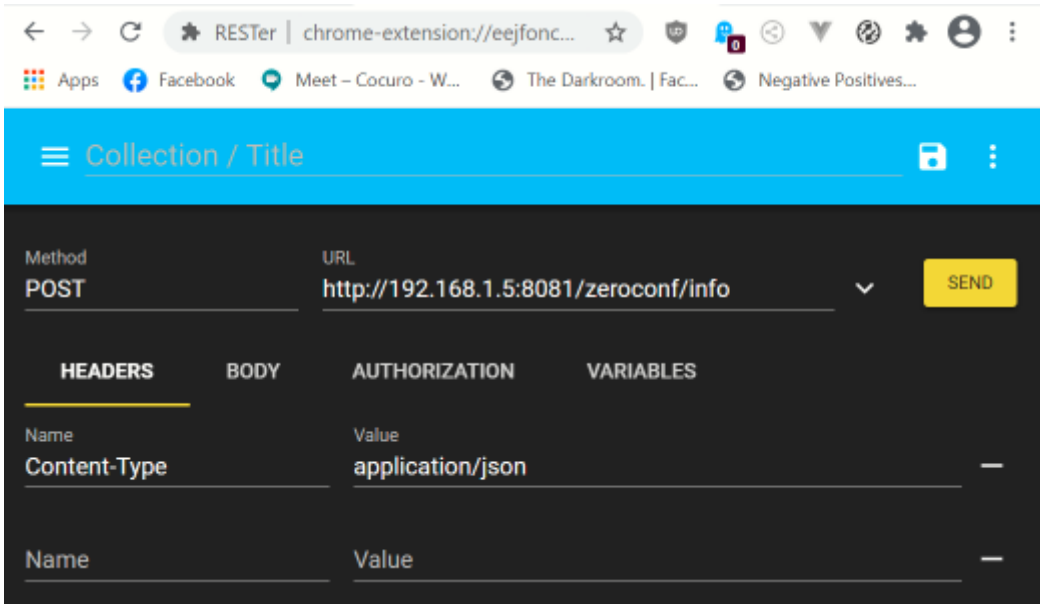
Find the new device's IP address. I suggest connecting your phone or tablet to the same network and using the Fing app.

> *Tip: On my network, the new device does not appear in my router's list of connected devices despite the fact that it got its IP address from the router's DNS service. I do not know if this is a "feature" of my router or a consequence of the mDNS protocol used by the Sonoff in DIY mode. In any event, connecting my phone to the same network and using the Fing app detected the Sonoff device. To confirm that you have found the correct device, select the device and use the "Find Open Ports" option and Fing should find a "blackice-icecap" user console on port 8081.*

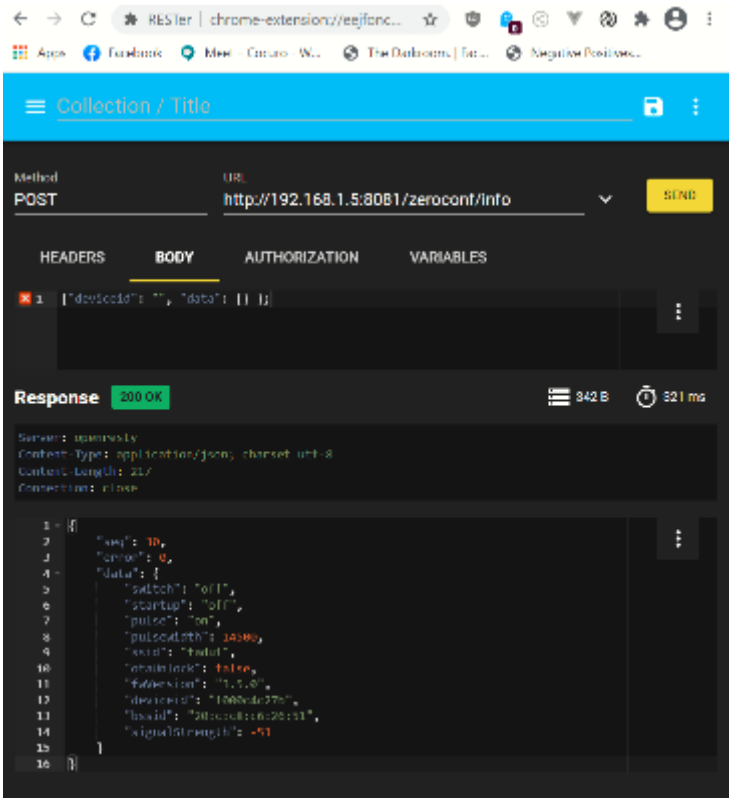
9) Testing RESTer and preparing to install the new firmware.

Open the RESTer plugin in Chrome and send the following command to request status information from the Sonoff where <IPADDR> is the Sonoff's IP Address that you just found. e.g. 192.168.1.5 in my case - yours will almost certainly be different.

Method: POST
URL: http://<IPADDR>:8081/zeroconf/info
Name: Content-Type
Value: application/json
Body: {"deviceid": "", "data":{}};



Click on the SEND button and you should get a response from the Sonoff device:



Note that the parameter "otaUnlock" has a value of "false". We will need to set this to "true" before we can send the new firmware.

Send the following command to change the value of "otaUnlock" to "true"

Method: POST
URL: http://<IPADDR>:8081/zeroconf/ota_unlock
Name: Content-Type
Value: application/json
Body: {"deviceid": "", "data":{}};

Now send the info command again to confirm that we have unlocked the OTA capability:

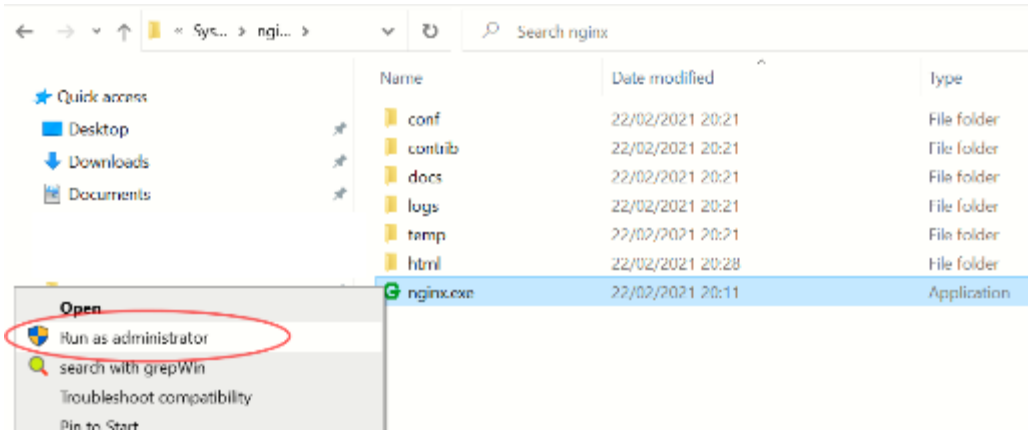
Method: POST
URL: http://<IPADDR>:8081/zeroconf/info
Name: Content-Type
Value: application/json
Body: {"deviceid": "", "data":{}};

This time, the value of "otaUnlock" must be "true" - if not, check your settings and try again.

10) Installing the new firmware.
If not already done so, download the nginx web server (http://nginx.org/en/download.html). Create a folder called nginx at the root level of the C: drive and extract the zip files into it. There is no installation: you run the nginx server by running the nginx.exe file as an Administrator.

Once unzipped and copied to the new directory, copy the new firmware's .bin file generated previously into the HTML subfolder.

Run the nginx software (nginx.bin in the new nginx folder) as Windows administrator. Verify that the file can be downloaded with a browser by entering the URL: http: //localhost/<filename>.bin (if nginx is running on a different PC, use the IP address instead of localhost (or if running on a different web server, use whatever URL is appropriate for that web server). Your browser should ask you to start to download the file if it finds the file OK - cancel the download.



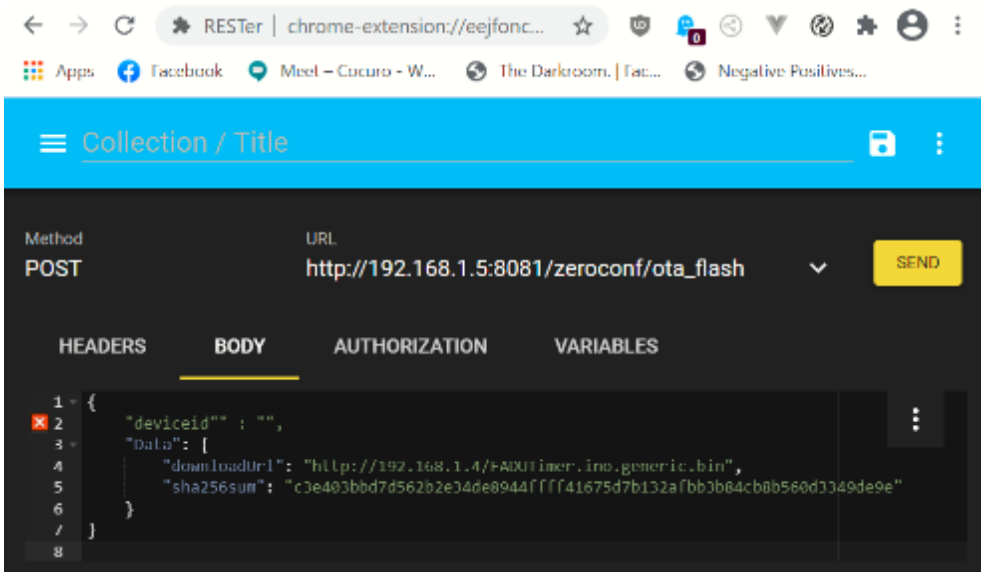
Obtain the SHA256 file checksum code for the new firmware by going to the page https://emn178.github.io/online-tools/sha256_checksum.html and dragging the .bin file there. This will be a long string of hexadecimal digits.

11) Aaaaaaand finally... Reprogram the Sonoff device's firmware with the following command:

Method: POST
URL: http://<IPADDR>:8081/zeroconf/ota_flash
Name: Content-Type
Value: application/json
Body:
{
"deviceid" : "",
"Data": {
"downloadUrl": "http://<WEB_IPADDR>/<filename>",
"sha256sum": "<SHA256Checksum>"
}
}

Where <WEB_IPADDR> is the IP Address of the web server, <filename> is the name of the .bin file and <SHAChecksum> is the checksum previously generated.

For my system, the Body looks like:



It will take a few seconds to upload the firmware and the LED on the Sonoff will display three fast flashes repeatedly while upload is in progress.

If the update was successful, the LED on the device stops flashing and turns off and the list of available Wi-Fi networks should now show one with SSID = FADUTimer (unless you renamed it in the firmware code before compiling it). Now you just have to follow the instructions elsewhere in the manual to link the timer app with the Sonoff device.

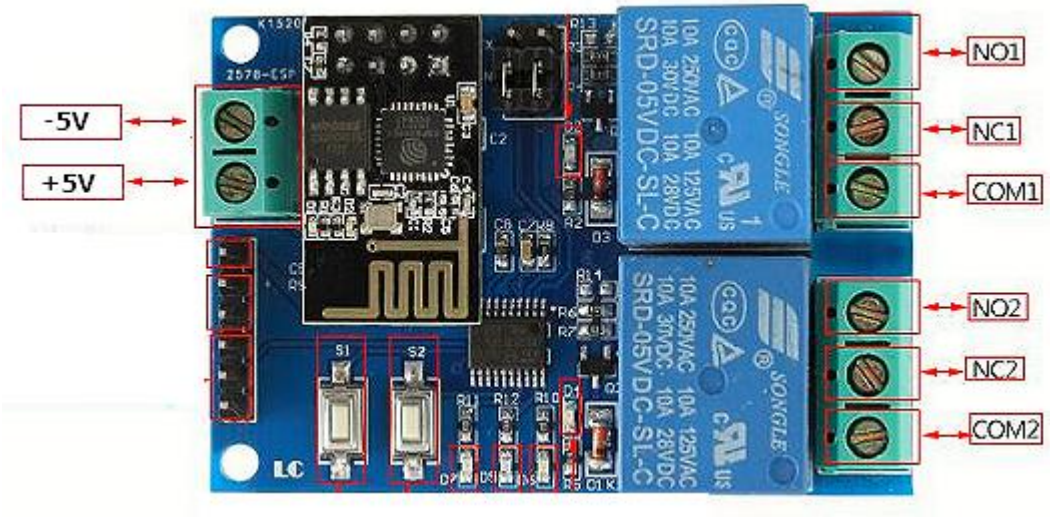
Tip: If the FADUTimer network does not show up, try restarting the Sonoff and see if it is visible on FADUTimer, or on its original ITEADxxxxxxxx network, or on your local network and try to figure out what went wrong. If it is on your local network, go through the instructions again starting at the section on Testing RESTer. Mine failed twice because I had put the wrong web server IP address in the POST command! When it failed, it simply sat there for several minutes in the "3 flashes" mode until I power-cycled it and it reconnected to my local network. When it succeeds in uploading the new firmware it only takes a handful of seconds.

Tip: If you are still having issues when you are sure you have updated the firmware OK (you can see the FADUTimer network) but the FADU timer app will not connect, use Fing on your phone/tablet to make sure you are on the correct (FADUTimer) network and that you have the correct IP address. Try using RESTer or Postman to drive the relay on the board using the REST API per the Sonoff documentation at: http://developers.sonoff.tech/basicr3-rfr3-mini-http-api.html.

Appendix 5: LC Technologies Relay Board

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 stabiliser.

On the plus side, it requires less electronics knowledge to configure – 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 interest are:

+5V	The positive side of the 5-volt power supply
-5V	The ground/earth side of the 5-volt power supply
COM1	The common terminal of Relay 1
NO1	The Normally Open terminal of Relay 1

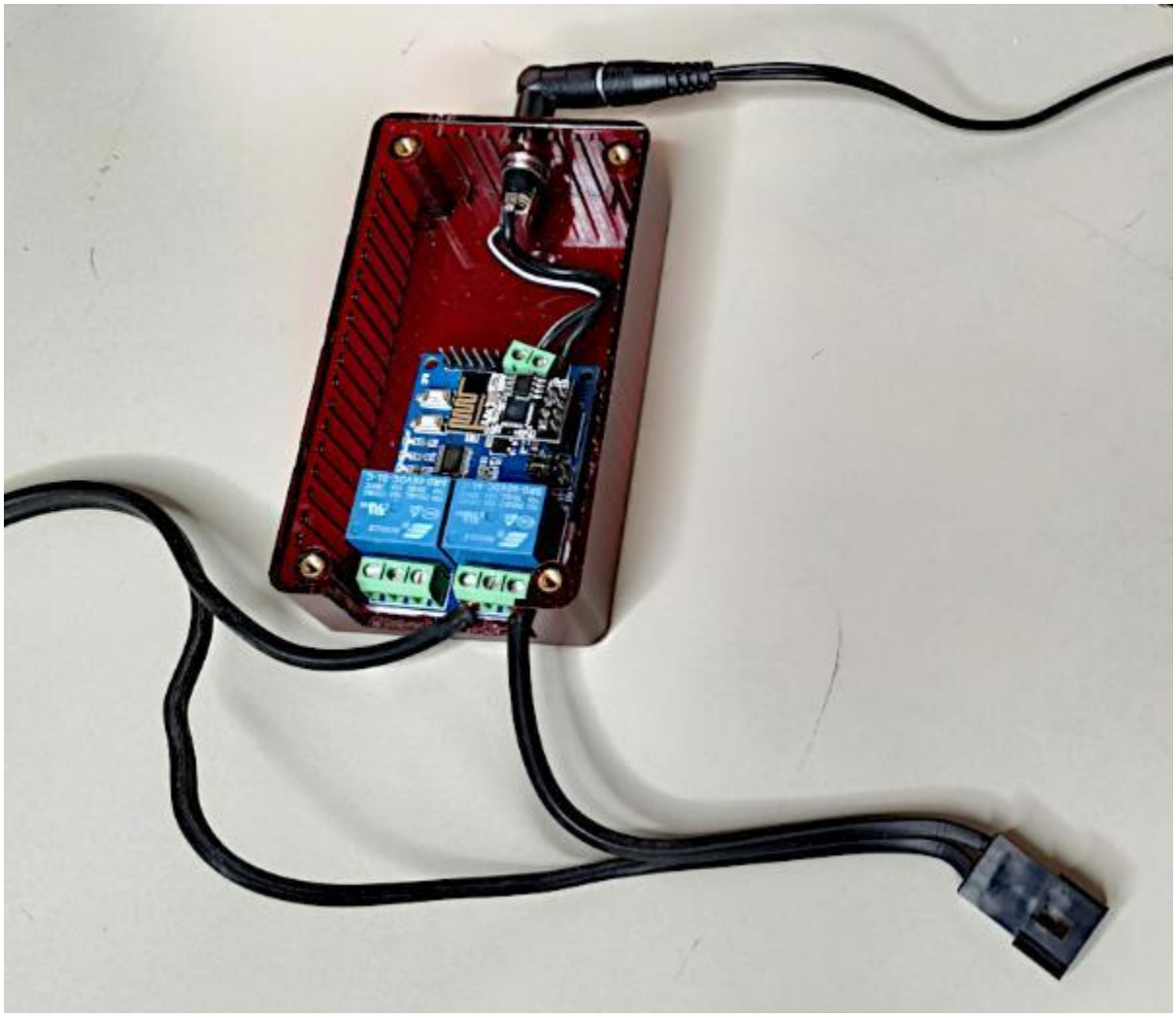
When the board is unpowered and when initially powered on, the NO (normally open) terminals are not connected to the COM (common) terminal of the relay. When the timer app tells the lamp to switch ON, the relay energises, and the NO terminal will connect to the COM terminal.

When the board is unpowered and when initially powered on, the NC (normally closed) terminals are connected to the COM (common) terminal of the respective relay. When the timer app tells the lamp to switch ON, the relay is energised, and this terminal is disconnected from the COM terminal.

Hence, to switch 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. The relay then simply acts as a switch in this lead. It does not matter which way around they are connected – the relay terminals connections make a simple mechanical switch. If your enlarger has other cables going to the enlarger, these may be for other features such as a fan operating at mains voltages so this module may not be suitable.

Regulated 5V dc power supplies are readily available from the usual online suppliers. Note that this *must* be a regulated 5 volts supply – a much 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 which is fine as long as you 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 chassis mounting 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 should use additional case-mounting connectors or strain-relief for the cables but this is just a proof-of-concept prototype.



I again advise against using this module with mains voltages, mainly due to the very small separation (1mm) between Relay-1's coil pin and the track running from the relay's COM pin to the terminal block. Relay-2's separation is even worse (!) but we are not using that one. Another consideration is that this is a very cheap module (with a very poor pcb layout) and if you believe the ratings on the relays then you are a more trusting person than I am.

Driving my LPL C7700 (via its voltage stabilizer!):

