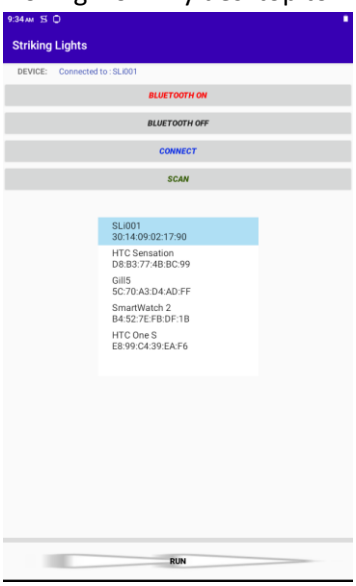
Programmable Digital LED controller – WS2811/12

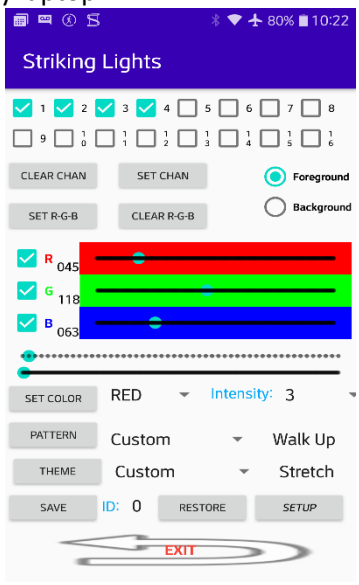
Android application StrikingLights

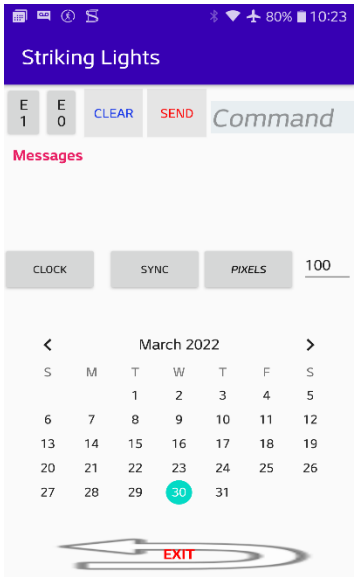
I learned Android Studio in about 3 to 4 months, enough to try to control my board using an old phone. I continued improving it but really it is not my goal to be an expert neither on JAVA nor Android. I'm an MSEE expert in FPGAs/ASICs and custom processors, hardware lab rat. I worked just enough to make the whole system to work. I would like Android/IOS experts to take the app and make it professional, right now it is a decent app but far to be delivered. Thanks to many developers that allow people like me to learn Android Studio, I used a lot of examples from the web.

The app can be tested without the hardware, I'm using another Android phone/tablet running a terminal emulator, and in this way you can connect and see the app layout, three frames working. This app works only with BT classic, it does not work with BT-LE, there are many examples how to do it, I have one but I'm not expending time working on that.

Any question about the Android app just let me know through email only fpgahelp@gmail.com , in this folder are two Android folders, A4 for phones that support Android 4.0 and up and A10 for tablets and support also Android 4 and up. The difference between A4 and A10 is the size of the text for phones and tablets; I don't know how deal with that so I created two different folders. If you download the app you must be knowledgeable with Android Studio and makes the necessary changes, I had some issues just moving from my desktop to my laptop.





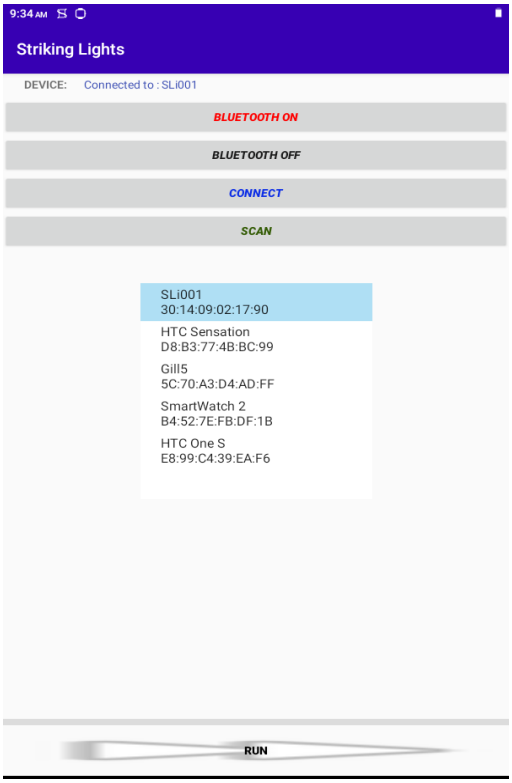


There are three frames; Frame1 is obvious, it is designed to turn on BT show pair devices and in this case connected to SLi001 hardware, also you can connect to TS7_EEA tablet running a terminal BT emulator. Frame2 is the application to control the LED strip, the commands and controls are explained in a doc in the ASM_compiler/docs folder. Frame3 send any command to the hardware and also is an attempt to implement a calendar where you can setup a special pattern/theme lights and colors for a time and day of the month like any calendar. When the time/day/month has an event it will send a sequence of commands similar to doing by hand. This feature is not implemented. (I don't know how to do it!).

To be able to run the Android app you need the hardware to connect, but there is a way to do it without a hardware using an Android phone/tablet running a Bluetooth terminal emulator. When the StrikingLights app connects to the emulator you will be able to see the three frames and the command sent by the app to the emulator. In the terminal emulator you can send any ASCI data back but there is

no need because the Android app does not verify any data back except help command, though you can display messages in Frame 3.

First frame:

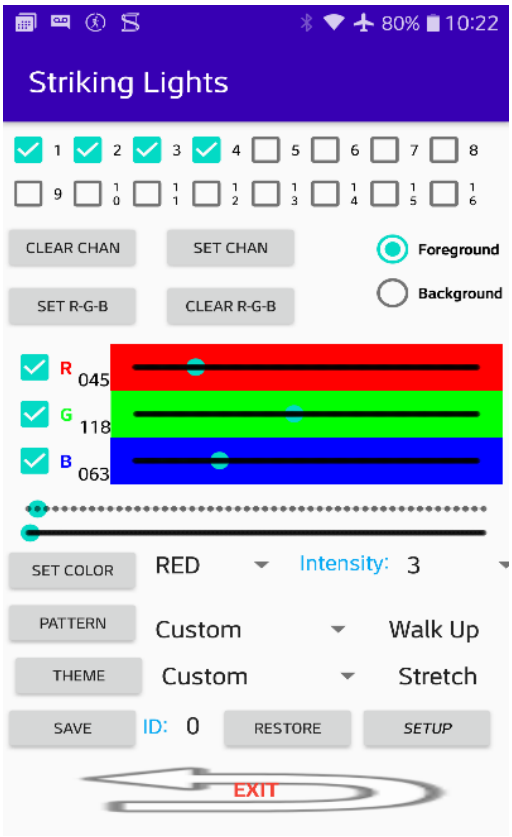


Bluetooth discovery and Bluetooth power ON and OFF, basic Android calls to manage Bluetooth devices.

SLi001 is the StrikingLights device 1, connect and disconnect feature nothing new here, the code only supports Bluetooth Classic.

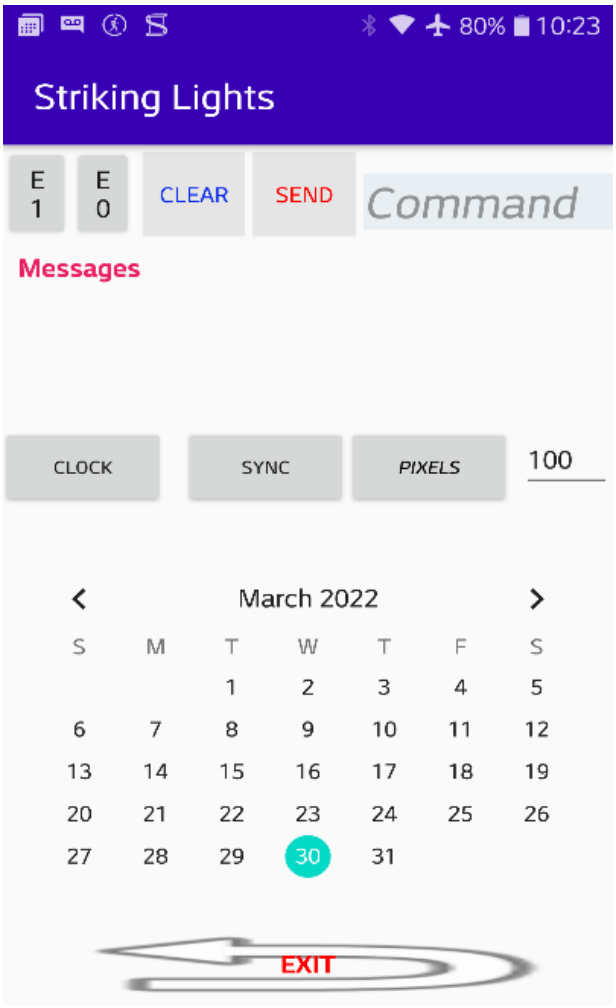
Clicking on Run button moves to next frame only if it is connected to a device

Second frame:



On top 16 channels from 1 to 16,
CLEAR CHAN unchecks all channels, ***SET CHAN*** checks all channels.
CLEAR R-G-B clear all colors all channels regardless of the channels check boxes.
SET R-G-B set colors defined by the color bar defined by the channel check boxes.
The three color seek bars ***R-G-B*** and the check boxes set the color for the channels.
The next two seek bars are timers used by especial effects, the first one sets the number of pixel to walk, the second one is for walking, blink and dimmer.
SET COLOR and ***Intensity*** set the color and intensity to all channels regardless of the channels check boxes, this 'solid color' feature works in conjunction with ***FOREGROUND*** and ***BAKGROUND***, selecting PATTERN in a mode different from Custom will use solid colors, basically solid colors is a quick way to set all channels to one of the 7 RGB colors, Custom uses the Theme and Pattern effects.

PATTERN and ***THEME*** are special features; ***THEME*** in Custom mode fills the buffer channel based on the colors in the number of colors in the queue, there are predefined themes in the selection and they are hard coded. The Theme mode selected either Stretch or Repeat affects the way the buffer is filled. Stretch mode fills the buffer evenly based on the number of colors in the color queue, for example if you click SET R-G-B or SET COLOR 3 times, with different colors, the LED strip will have three colors evenly distributed along the length of the strip. Repeat mode will take the three colors and repeat as many times it will fit in the strip length defined by Pixel Numbers (set in Frame 3).
PATTERN really is pattern effects, two types of pattern effects, Custom and predefined, Custom uses pattern mode to display the effects like walk up, down up and down and blink. Non Custom is only for solid colors based on foreground and background colors. This is more visible running the application in hardware.
SAVE it saves the current selected channels from local memory to external memory, the unchecked channels are unmodified. ID is the file descriptor/filename for the set of channels saved.
RESTORE it restore the selected channels from external memory to local memory. ID is the file descriptor/filename for the set of channels saved.
ID is basically the offset of the 16 channels in the external memory, only PAGE 0 is supported in this design. A PAGE number can be added to make more room for more saves.
SETUP goes to the next Frame.



E1 is Echo ON command and ***E0*** is Echo OFF, enables and disables the hardware sending messages back.
CLEAR deletes the display messages
SEND it sends to the hardware the characters in the command field.
MESSAGES, the field to display messages from the hardware.
Calendar is an attempt to send commands to the hardware based on events. It is not implemented.
EXIT always return to previous menu

The following tables are the single ASCII commands :

```
------Main Menu------
 a = application
 s = system setup
 d = debug
 v = version
 L = Load Program *.mem
 E = Echo ON
 e = echo OFF
 C = Clear Screen ON
 c = clear screen OFF
 I = IRQ mask
```

```
------Application------
 c = set channel (0 to f)
 f = select foreground
 b = select background
```

s = **s**et color value per channel
p = **p**attern
e = th**e**me
d = **d**isplay theme
m = theme **m**ode ----------→stretch, repeat
i = **i**ntensity (0-7)
0 = blk
1 = red
2 = grn
3 = yel
4 = blu
5 = mag
6 = cya
7 = wht
8 = update colorset queue
9 = reset colorset queue
a = **a**ctive pixels ----------→ number of pixels to walk
r = **r**ead buffer mode ----------→ walk up/walk dn/walk ud/blink
t = **t**imer (blink/dim/walk) --------→ general timer for effects
j = dimm steps ----------→ simple counter to dim channels
n = **n**umber of Pixels
S = **S**ave channel to virtual channel
R = **R**estore virtual channel to channel
P = external memory **P**age
V = set **V**irtual channel, V=-1 then V=c
k = display cloc**k** ----------→ real time clock
v = **v**ersion
h = **h**elp
x = e**x**it ----------→ return to Main Menu

------System Setup------

t = **t**rigger source (0 to 3)
s = **s**cope Source (0 to 3) ----------→ lower case
l = PWM reset ----------→ letter l
p = PWM period
0 = PWM pulse width code 0
1 = PWM pulse width code 1
8 = **8**00 KHz PWM
4 = **4**00 KHz PWM
b = set **b**ase reg address
r = **r**ead all registers ----------→ lower case
w = **w**rite/read port/register --------→ lower case
n = **n**umber of Pixels
Y = **Y**ear
M = **M**onth
D = day/**D**ate
H = **H**our
U = min**U**te
S = **S**ec ----------→ upper case
k = display cloc**k**
R = target **r**eset ----------→ upper case
W = **W**rite protect (2 or 0)----------→ upper case
v = display **v**ariables
x = e**x**it

------ Debug Menu------

A = memory **A**ccess mode
B = memory **B**yte mode
P = external memory **P**age
m = **m**emory write/read
d = Memory **d**ump
n = **n**ext dump
f = **f**ill memory address/value (aaaa,vvvv)
l = set **l**ength/increment (llll,iiii)
w = **w**rite/read port/register
s = **s**et reg bit number
c = **c**lear reg bit number
t = **t**oggle reg bit number
i = Pulse reg b**i**t number
a = **a**nd
o = **o**r
e = xor
b = set **b**ase I/O/Registers address
r = **r**ead all Input ports/registers
v = display **v**ariables
h = **h**elp
x = e**x**it
