Twitter Thread by foone

foone
@Foone

I went looking for a remote-controlled power switch (the wireless christmas kind, not the modern IoT kind) and didn't find it, but I did find this thing I bought just to figure out why it exists.
It's a timer outlet, but you program it from your phone... but it's not wireless.

Yeah instead it plugs into your phone's HEADPHONE PORT

I got it on sale.
probably because iphone dropped the headphone port and they had to get with the 21st century and make it bluetooth

So inside the box is the device. it's got a little 2-prong polarized outlet with no ground.

on top is the only controls: on/off/timer.

then there's a 3.5mm jack here

turns out there's one more button! it's a reset button.
I guess the thing saves settings when turned off, because you have to unplug it to push the reset button.
Specs: up to 10 amps for a resistive load, and up to 5 amps for a tungsten load.

and this is just an aux cable

I hate when they're just like "download the app by searching for foobar".
That's putting some serious trust in your SEO, man

this is what the app is supposed to look like

thankfully it's still on the app store.
although it tries to sell me a bunch of unrelated movies first?

uh-oh, 2 stars?

one of my favorite things to do is to look up the ratings on IoT apps... they're never good.

apparently it requires a lot of permissions and barely works

huh, triangle screws.
whelp, guess I better get my dremel and paper clips!

so the app wants to take pictures, access your location,
make calls, and access all your files.
and if you deny it, it just dumps you in the settings page to fix permissions, with no message.

huh, not all of the screws are triangles.
there's 4 of them, and 2 are philips

I hoped that'd make more sense when I opened it up, like one went into a PCB and the other didn't, but NOPE! it's just because Reasons.

So here's the control board.
We've got a CPU and two smaller chips. Probably one is some kind of communication chip, and the other is a flash chip for storing settings?

Nothing on the other side but the LED.

Although this bit is interesting: L/R/+/-, on the cables going to the other board.
L isn't connected... I think that means there's a version of this that can control two outlets at once, not just one.

also HIDDEN BATTERY!
someday that will die and leak and the whole thing will be destroyed.

The other PCB.
I do like that they keep all the high-voltage AC stuff separate from the low-voltage DC stuff.
Cheaper versions of this would have just had one PCB.

That big box is a Massuse ME-11-I-012-1HS3AF relay.

So that's an inrush type relay, 12v coil voltage,1A contact form, sealed, 16amp rating, AgSnO■ contacts, class-F insulation.

So back on the main PCB, let's look at that CPU.
It's a Sino Wealth SH79F166A.

which is an 8-bit microcontroller with 16 kilobytes of flash ROM, 256 bytes of RAM, and 1 kilobyte of eeprom-like storage.

AND IT'S AN 8051! EVERYONE TAKE A DRINK

Over here is a Holtek HT9274.
That's a quad-op-amp.

And this is a 026B-A-CF850S, which is an... air filter? hmm.

actually it turns out it's a battery charger/management chip, an XT2051.
because it has a battery, yeah.

so I was wrong, it does all the storage inside the chip itself! fancy.

so apparently the communication with the phone/tablet is two way!
because it can tell it's not connected properly, in this emulator I'm using

why does this look so iOS
it's an android app

thankfully they didn't obfuscate their java code, so I can see sound generation code.
it sounds like (NO PUN INTENDED) it has a protocol of simple tones that it plays at the device.

it also might not be two way:
android historically has had a AudioManager.isWiredHeadsetOn api which tells you if the 3.5mm jack is connected.
So it may just be detecting there's no headphones plugged in to my emulator.

sadly since it's gaming oriented it doesn't seem to have any way to shim that out.

so the Toner class has a bunch of methods that do various things, like playOn to turn it on, playRandom, playProgram, and playDusk (and "dust"? they seem to mix up dust and dusk a lot)

so to turn it on you send the simple command "1111".

and we can see over in getSimpleCommand that a simple command is a command + a clock sync + a length, then there's a checksum. And it logs all this for us! handy.

the clock sync stuff is the current date, daylights savings times, timezone, latitude, longitude, then CT and CD.
CT is "current time" as an integer of how many minutes it is into the day, and CD is the day of the week.

the day of the week is implemented in binary, with a fallback in case you're on an INVALID DAY.
(it's using Monday = 001, and counting up from there)

and here's the checksum function.
uhhh. I'm not sure I'm awake enough to figure this out, but... it starts by padding up to a multiple of 8 bits.

then it calculates a total sum by converting every group of 8 bits to an integer and adding them together
then it converts that to a binary number, and pads it out (on the left this time) to 8 bits

then it chops the checksum down to 8 bits... and checks if the last digit is a 1.
if it is, it adds a 1?

it's doing something like count up the bits, add that sum, but then add an extra 1 if it was odd. I think that means it's different lengths for even or odd? I may just have to stick this code in a harness and run it

yeah. it is variable length.
I don't know if that was intentional. I kinda don't think so.