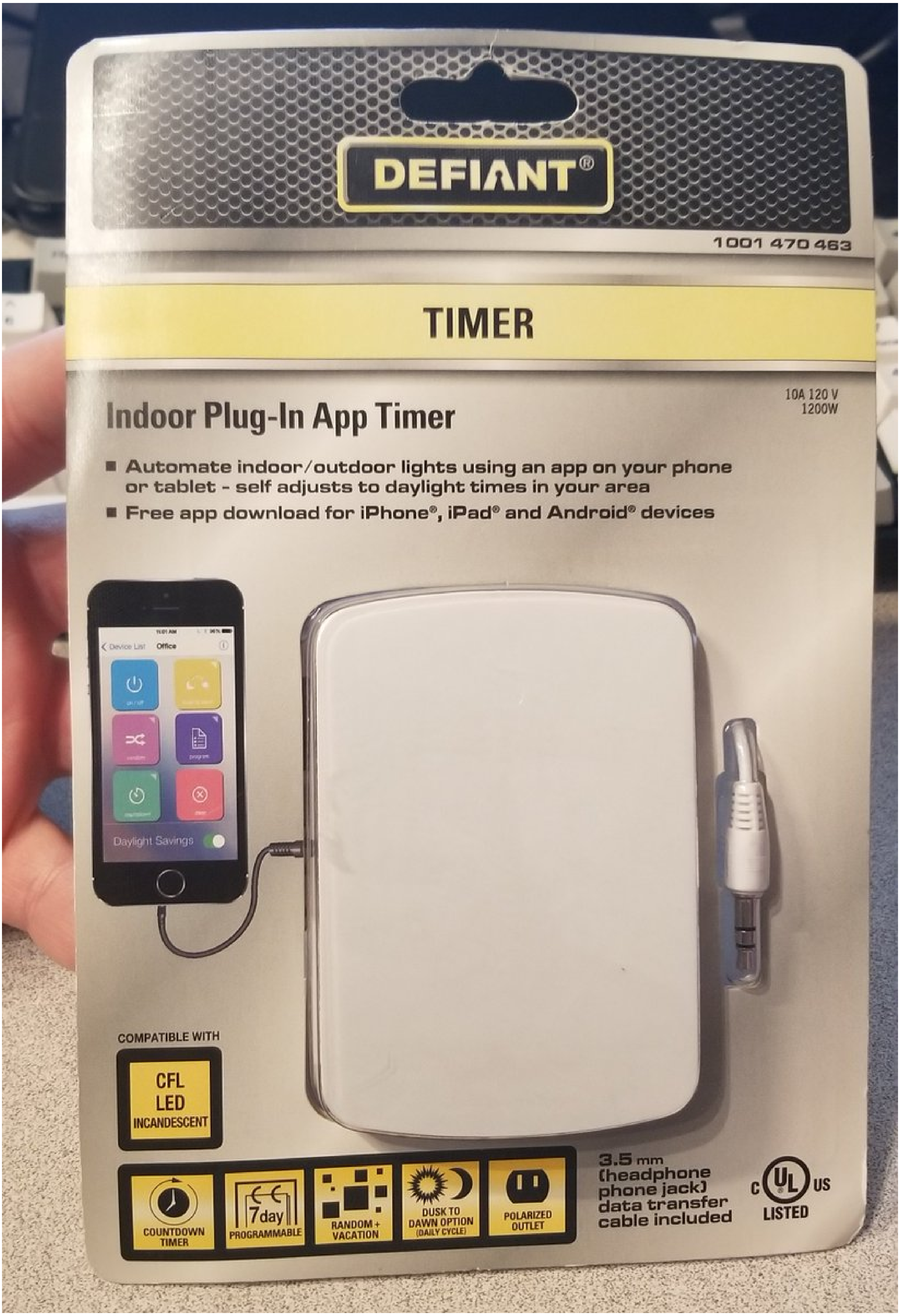
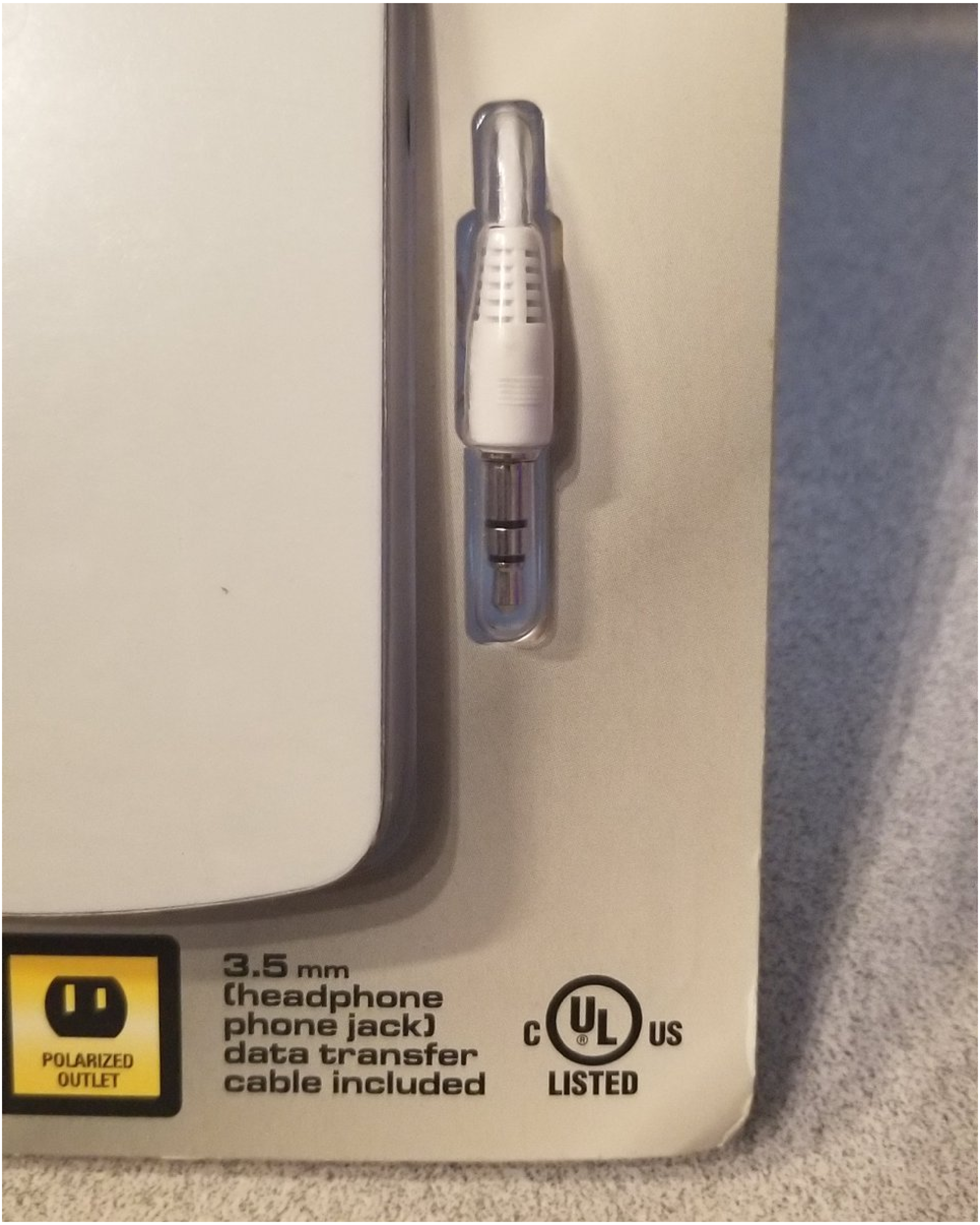
Twitter Thread by foone

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.

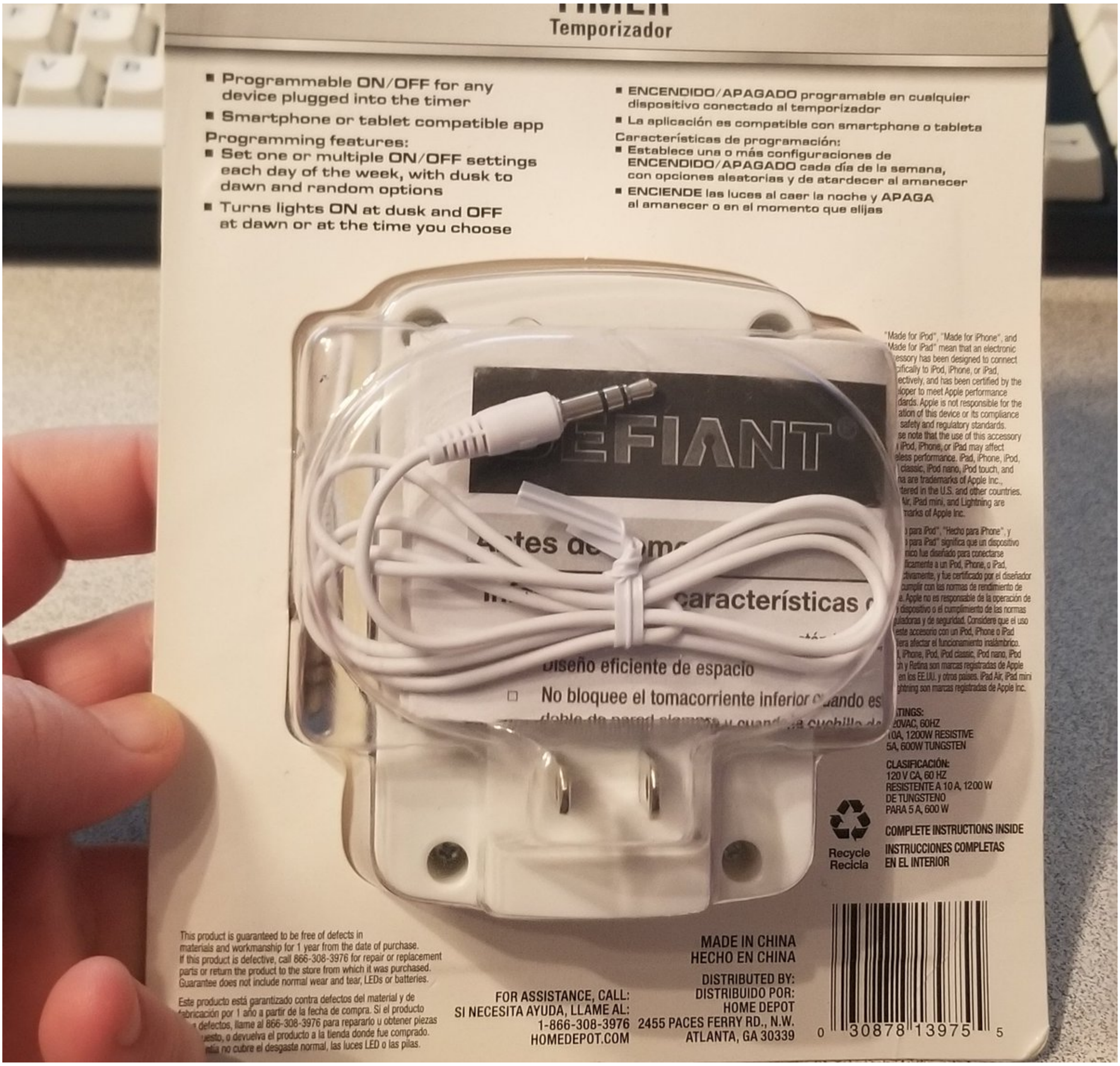

DEFIANT®
1001 470 463
TIMER
Indoor Plug-In App Timer
10A 120 V
1200W
Automate indoor/outdoor lights using an app on your phone or tablet - self adjusts to daylight times in your area
Free app download for iPhone®, iPad® and Android® devices
Daylight Savings
COMPATIBLE WITH
CFL
LED
INCANDESCENT
COUNTDOWN TIMER
7day
PROGRAMMABLE
RANDOM + VACATION
DUSK TO DAWN OPTION (DAILY CYCLE)
POLARIZED OUTLET
3.5 mm (headphone phone jack) data transfer cable included
c UL us
LISTED

Yeah instead it plugs into your phone's HEADPHONE PORT

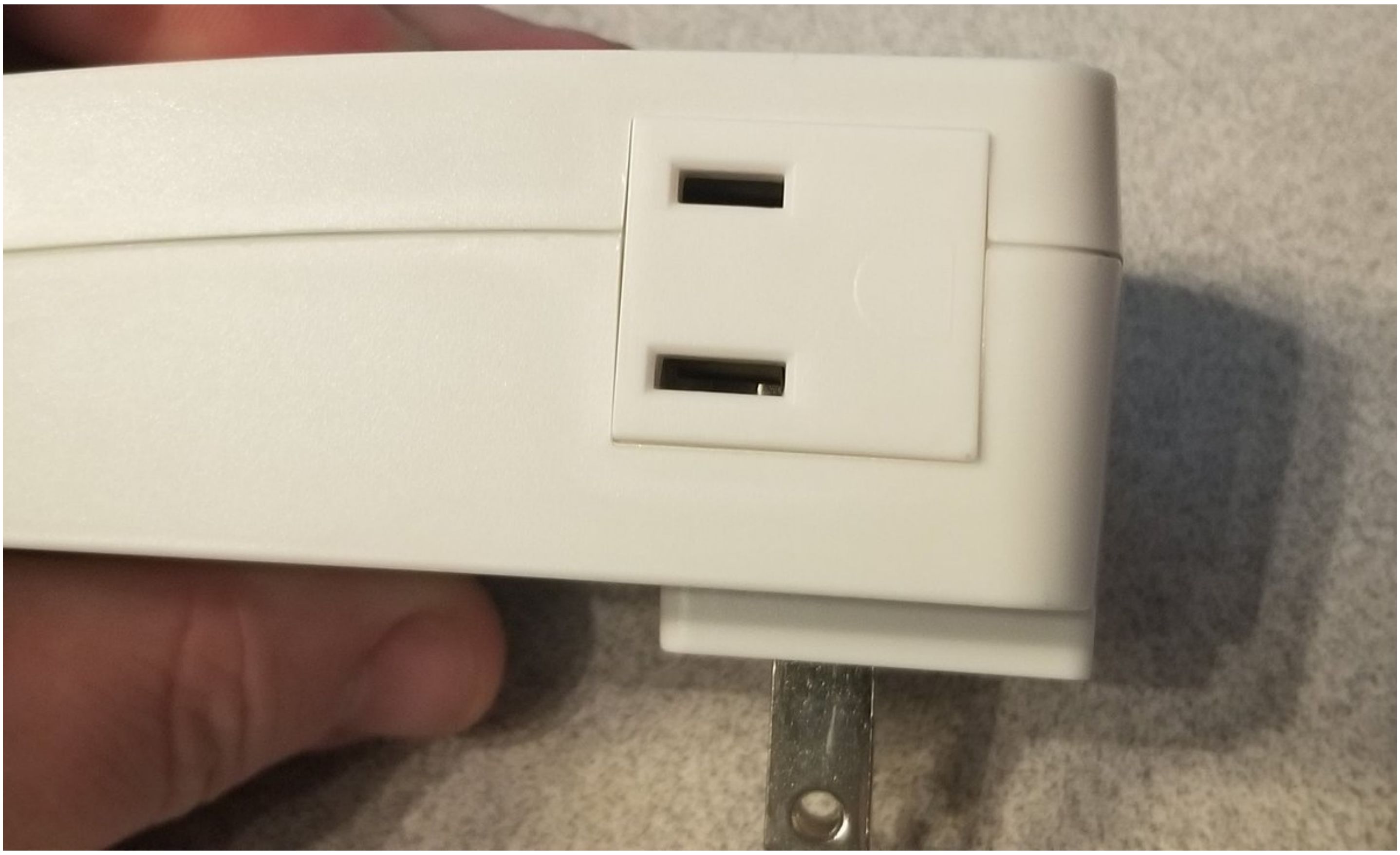


I got it on sale.

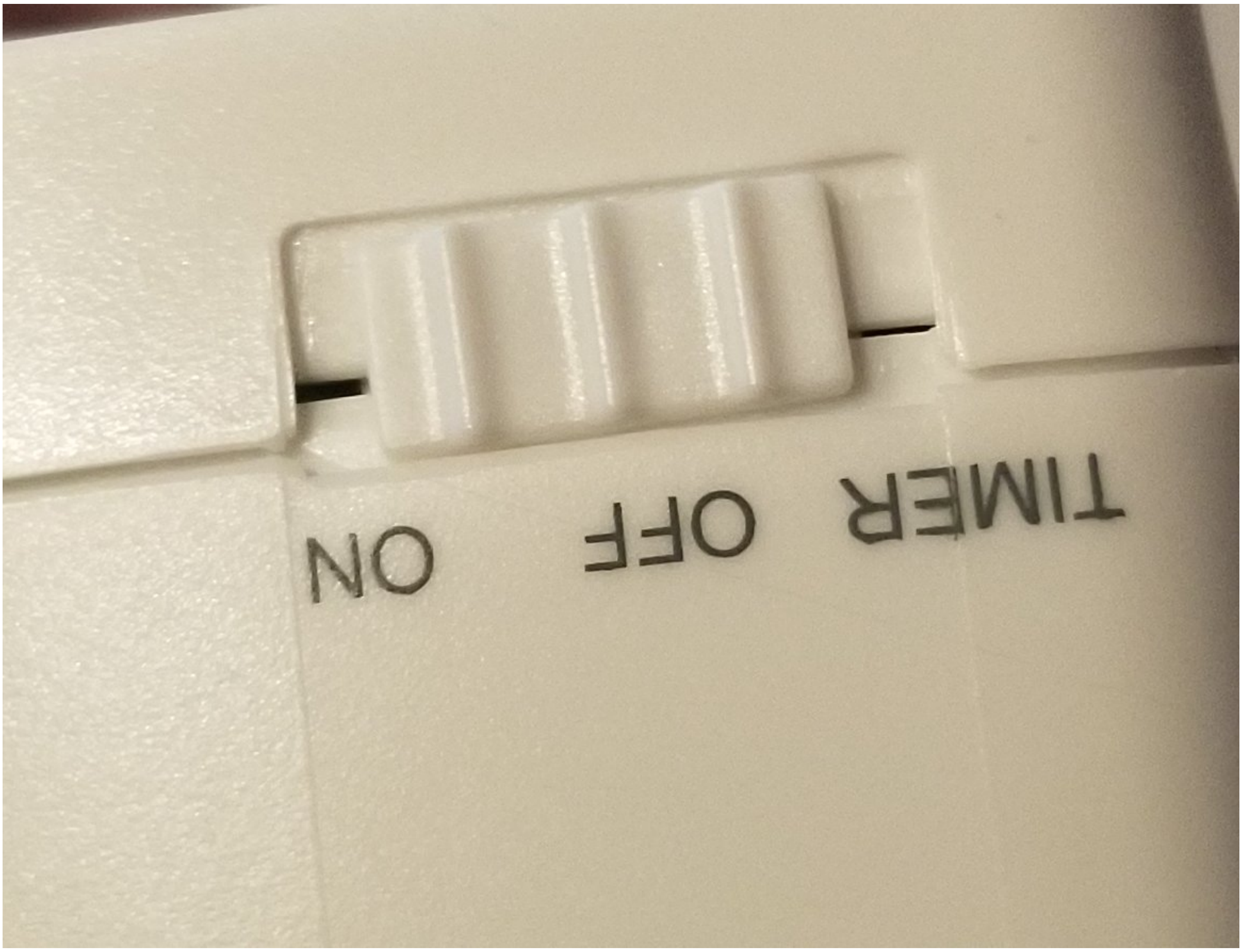
probably because iphone dropped the headphone port and they had to get with the 21st century and make it bluetooth


Temporizador
Programmable ON/OFF for any device plugged into the timer
Smartphone or tablet compatible app
Programming features:
Set one or multiple ON/OFF settings each day of the week, with dusk to dawn and random options
Turns lights ON at dusk and OFF at dawn or at the time you choose
ENCENDIDO/APAGADO programable en cualquier dispositivo conectado al temporizador
La aplicación es compatible con smartphone o tableta
Características de programación:
Establece una o más configuraciones de ENCENDIDO/APAGADO cada día de la semana, con opciones aleatorias y de atardecer al amanecer
ENCIENDE las luces al caer la noche y APAGA al amanecer o en el momento que elijas
DEFIANT
Diseño eficiente de espacio
No bloquee el tomacorriente inferior cuando es
RATINGS: 120VAC, 60HZ 10A, 1200W RESISTIVE 5A, 600W TUNGSTEN
CLASIFICACIÓN: 120 V CA, 60 HZ RESISTENTE A 10 A, 1200 W DE TUNGSTENO PARA 5 A, 600 W
Recycle Recicla
COMPLETE INSTRUCTIONS INSIDE
INSTRUCCIONES COMPLETAS EN EL INTERIOR
This product is guaranteed to be free of defects in materials and workmanship for 1 year from the date of purchase. If this product is defective, call 866-308-3976 for repair or replacement parts or return the product to the store from which it was purchased. Guarantee does not include normal wear and tear, LEDs or batteries.
FOR ASSISTANCE, CALL: SI NECESITA AYUDA, LLAME AL: 1-866-308-3976 HOMEDEPOT.COM
MADE IN CHINA HECHO EN CHINA
DISTRIBUTED BY: DISTRIBUIDO POR: HOME DEPOT 2455 PACES FERRY RD., N.W. ATLANTA, GA 30339
0 30878 13975 5

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.

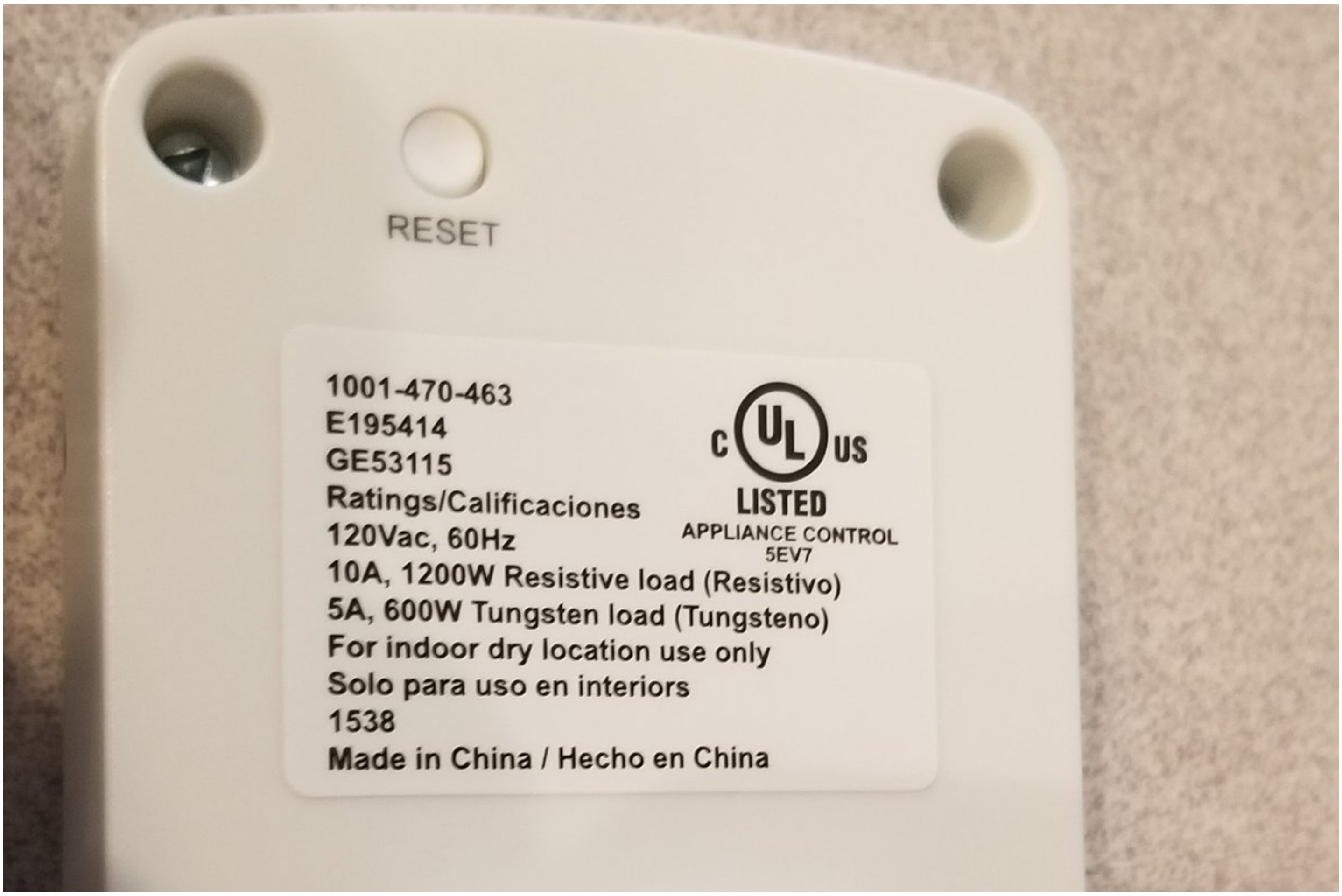


then there's a 3.5mm jack here

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.

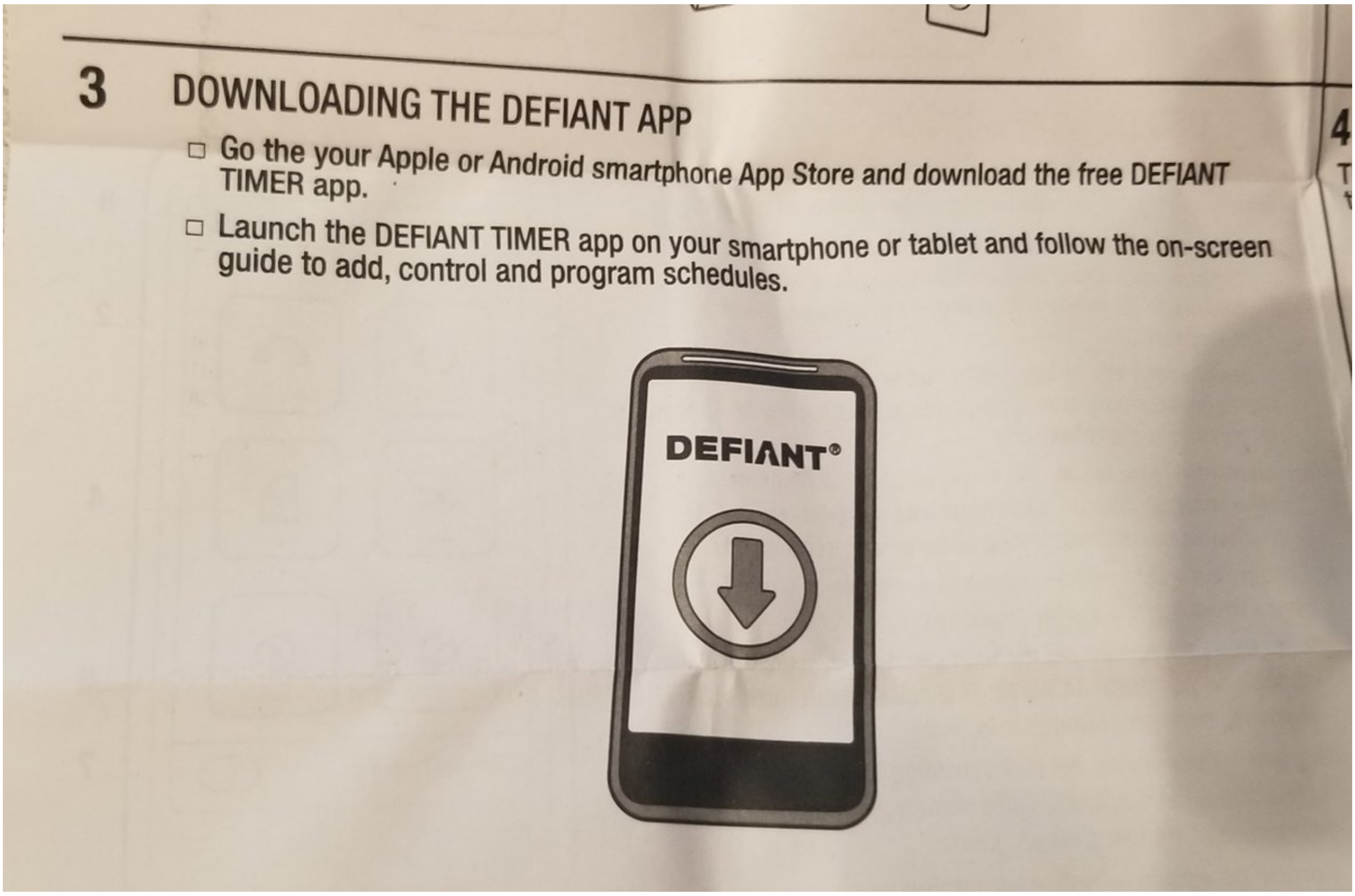


1001-470-463
E195414
GE53115
Ratings/Calificaciones
120Vac, 60Hz
10A, 1200W Resistive load (Resistivo)
5A, 600W Tungsten load (Tungsteno)
For indoor dry location use only
Solo para uso en interiors
1538
Made in China / Hecho en China

C UL US
LISTED
APPLIANCE CONTROL
5EV7

and this is just an aux cable

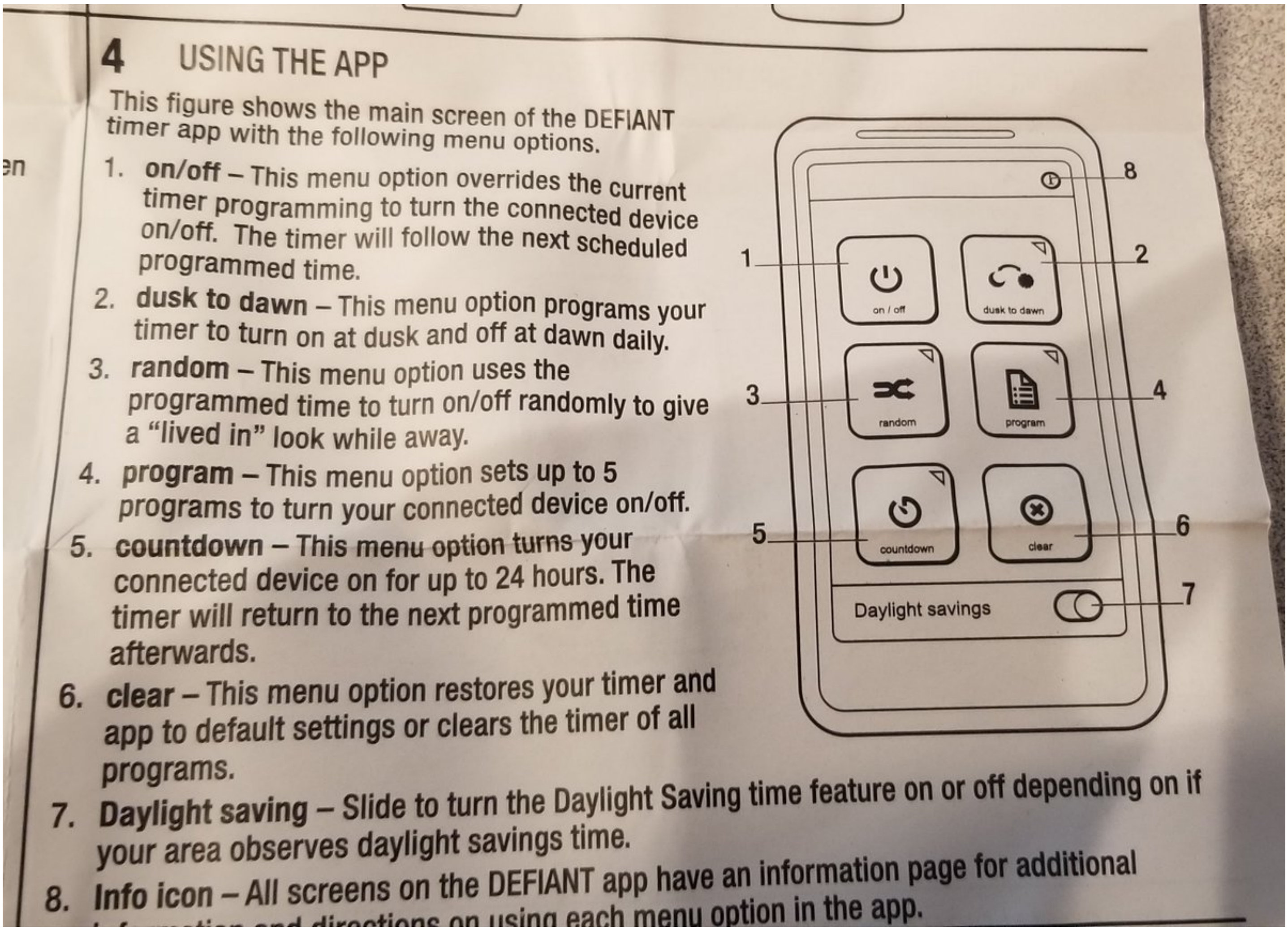
I hate when they're just like "download the app by searching for foobar".
That's putting some serious trust in your SEO, man



3 DOWNLOADING THE DEFIANT APP

- □ Go the your Apple or Android smartphone App Store and download the free DEFIANT TIMER app.
- □ Launch the DEFIANT TIMER app on your smartphone or tablet and follow the on-screen guide to add, control and program schedules.

this is what the app is supposed to look like



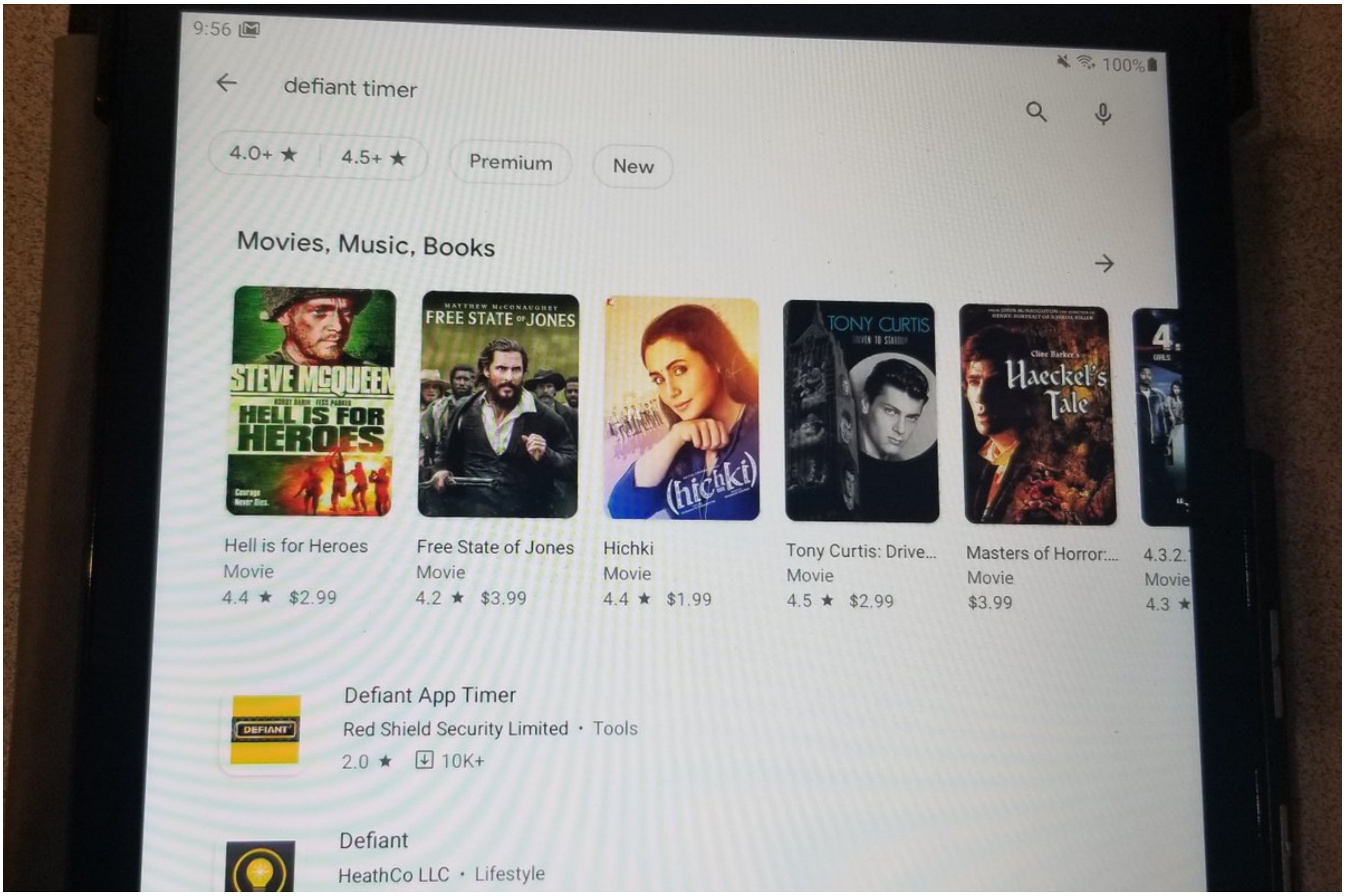
4 USING THE APP

This figure shows the main screen of the DEFIANT timer app with the following menu options.

1. **on/off** – This menu option overrides the current timer programming to turn the connected device on/off. The timer will follow the next scheduled programmed time.
2. **dusk to dawn** – This menu option programs your timer to turn on at dusk and off at dawn daily.
3. **random** – This menu option uses the programmed time to turn on/off randomly to give a "lived in" look while away.
4. **program** – This menu option sets up to 5 programs to turn your connected device on/off.
5. **countdown** – This menu option turns your connected device on for up to 24 hours. The timer will return to the next programmed time afterwards.
6. **clear** – This menu option restores your timer and app to default settings or clears the timer of all programs.
7. **Daylight saving** – Slide to turn the Daylight Saving time feature on or off depending on if your area observes daylight savings time.
8. **Info icon** – All screens on the DEFIANT app have an information page for additional information and directions on using each menu option in the app.

thankfully it's still on the app store.

although it tries to sell me a bunch of unrelated movies first?



Movies, Music, Books

Hell is for Heroes
Movie
4.4 ★ $2.99

Free State of Jones
Movie
4.2 ★ $3.99

Hichki
Movie
4.4 ★ $1.99

Tony Curtis: Drive...
Movie
4.5 ★ $2.99

Masters of Horror:...
Movie
$3.99

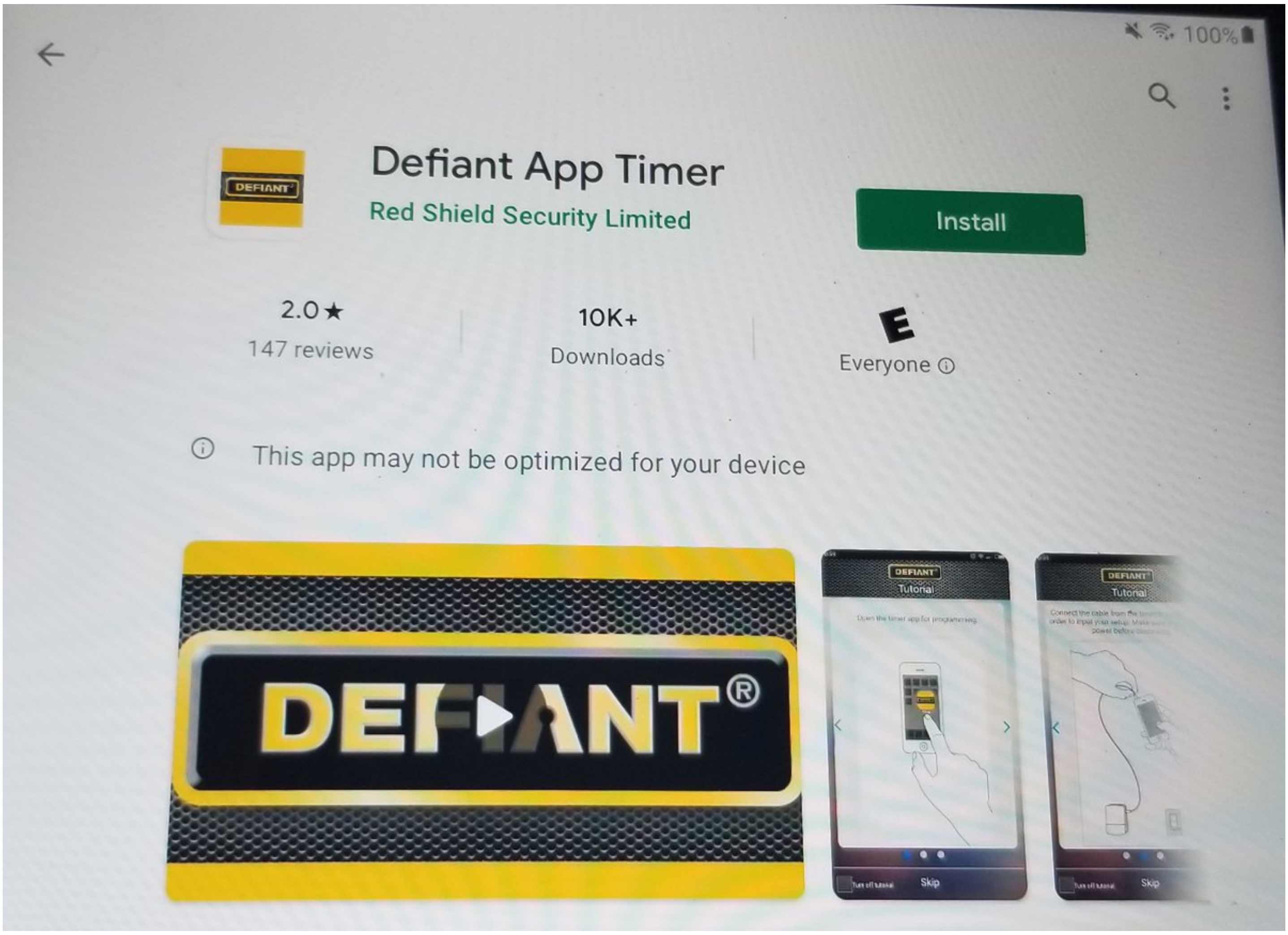
4.3.2.
Movie
4.3 ★

Defiant App Timer
Red Shield Security Limited · Tools
2.0 ★ 10K+

Defiant
HeathCo LLC · Lifestyle

uh-oh, 2 stars?

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



Defiant App Timer

Red Shield Security Limited

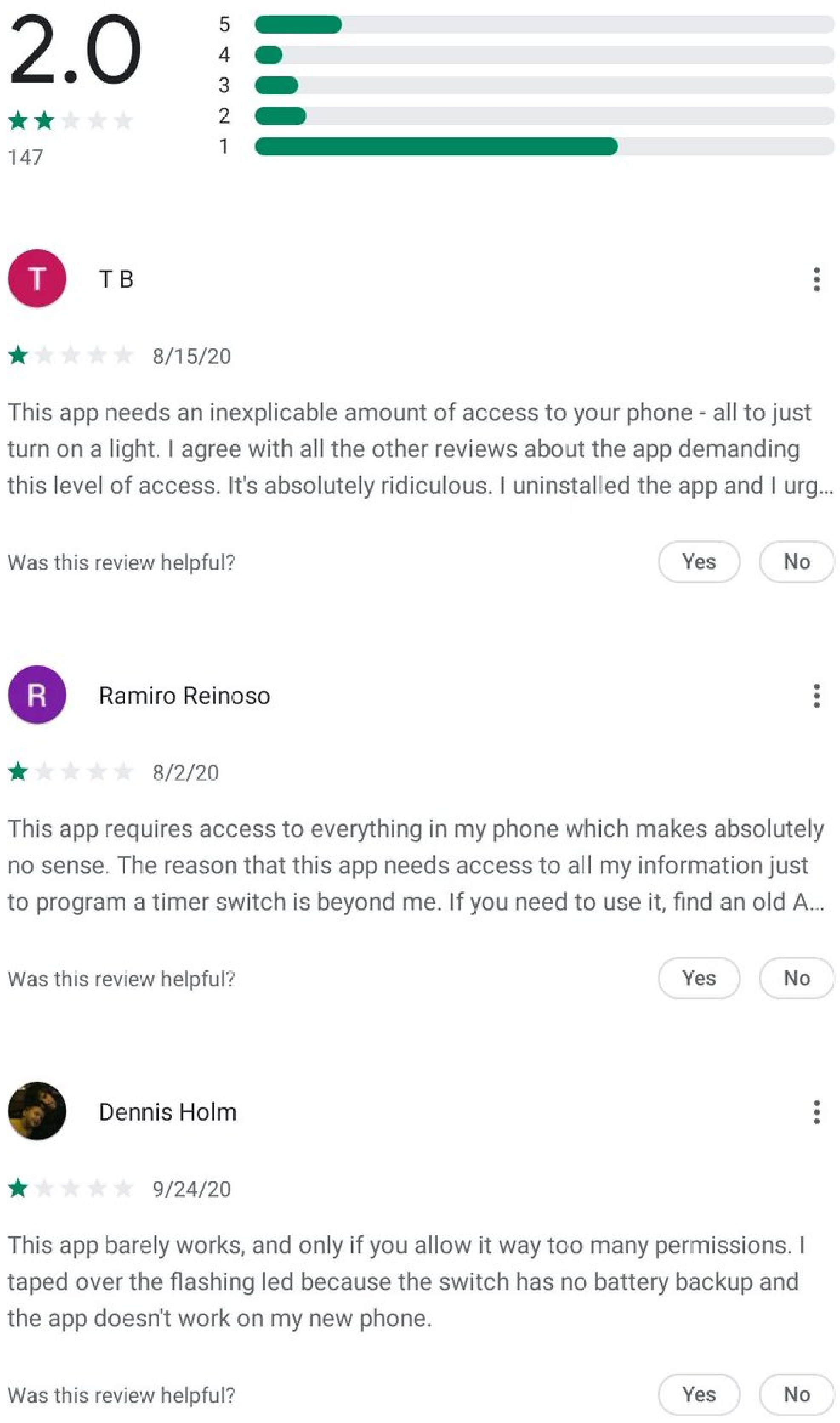
2.0★ 147 reviews

10K+ Downloads

Everyone

This app may not be optimized for your device

apparently it requires a lot of permissions and barely works



2.0

★★☆☆☆

147

5
4
3
2
1

T B

★☆☆☆☆ 8/15/20

This app needs an inexplicable amount of access to your phone - all to just turn on a light. I agree with all the other reviews about the app demanding this level of access. It's absolutely ridiculous. I uninstalled the app and I urg...

Was this review helpful? Yes No

Ramiro Reinoso

★☆☆☆☆ 8/2/20

This app requires access to everything in my phone which makes absolutely no sense. The reason that this app needs access to all my information just to program a timer switch is beyond me. If you need to use it, find an old A...

Was this review helpful? Yes No

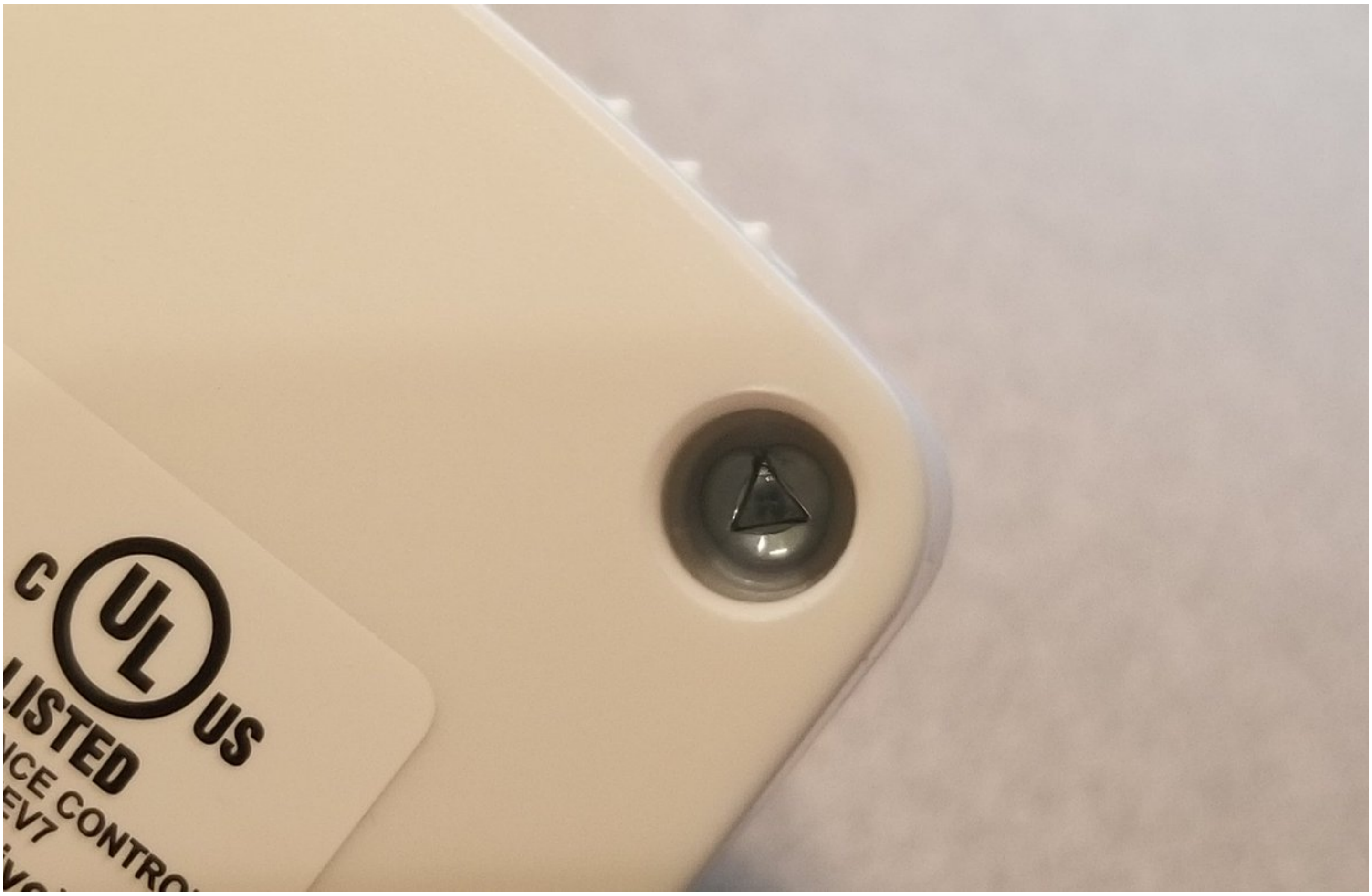
Dennis Holm

★☆☆☆☆ 9/24/20

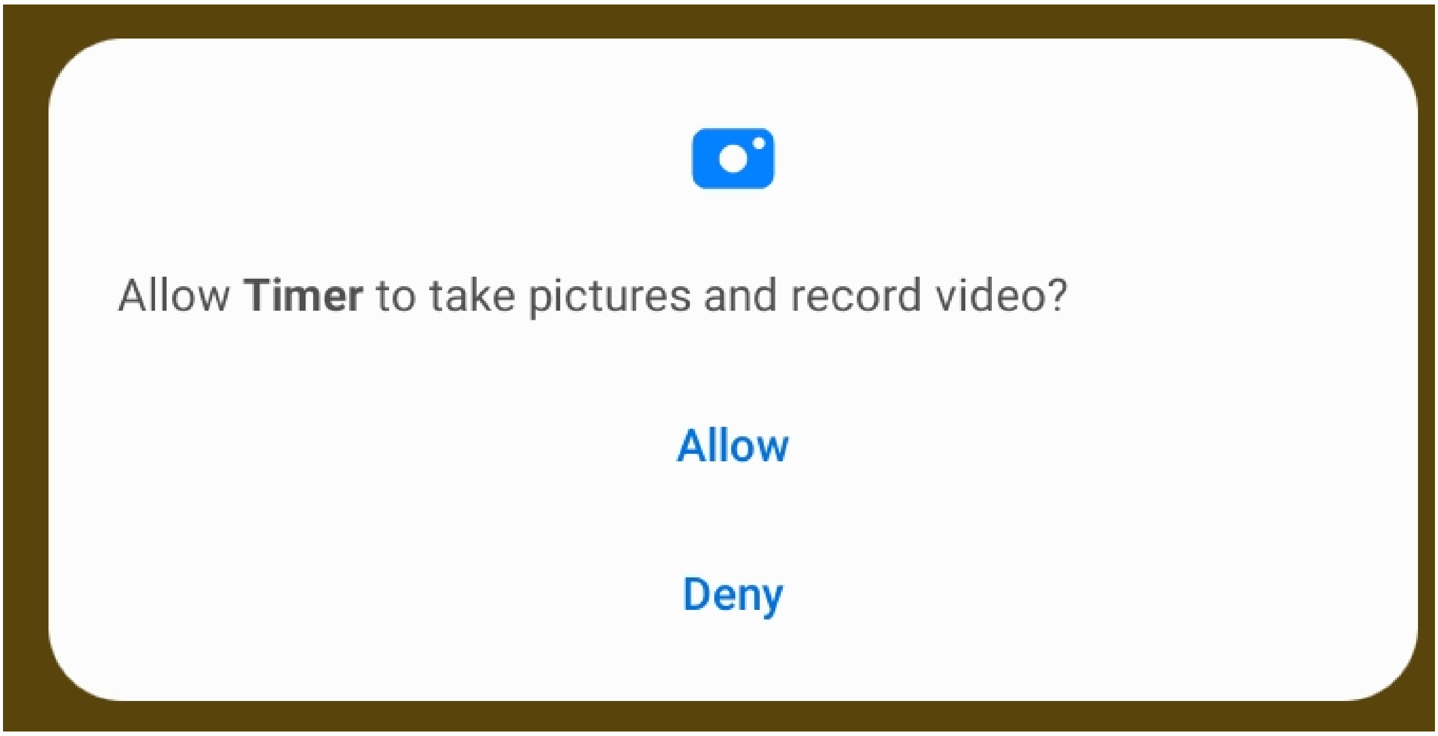
This app barely works, and only if you allow it way too many permissions. I taped over the flashing led because the switch has no battery backup and the app doesn't work on my new phone.

Was this review helpful? Yes No

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

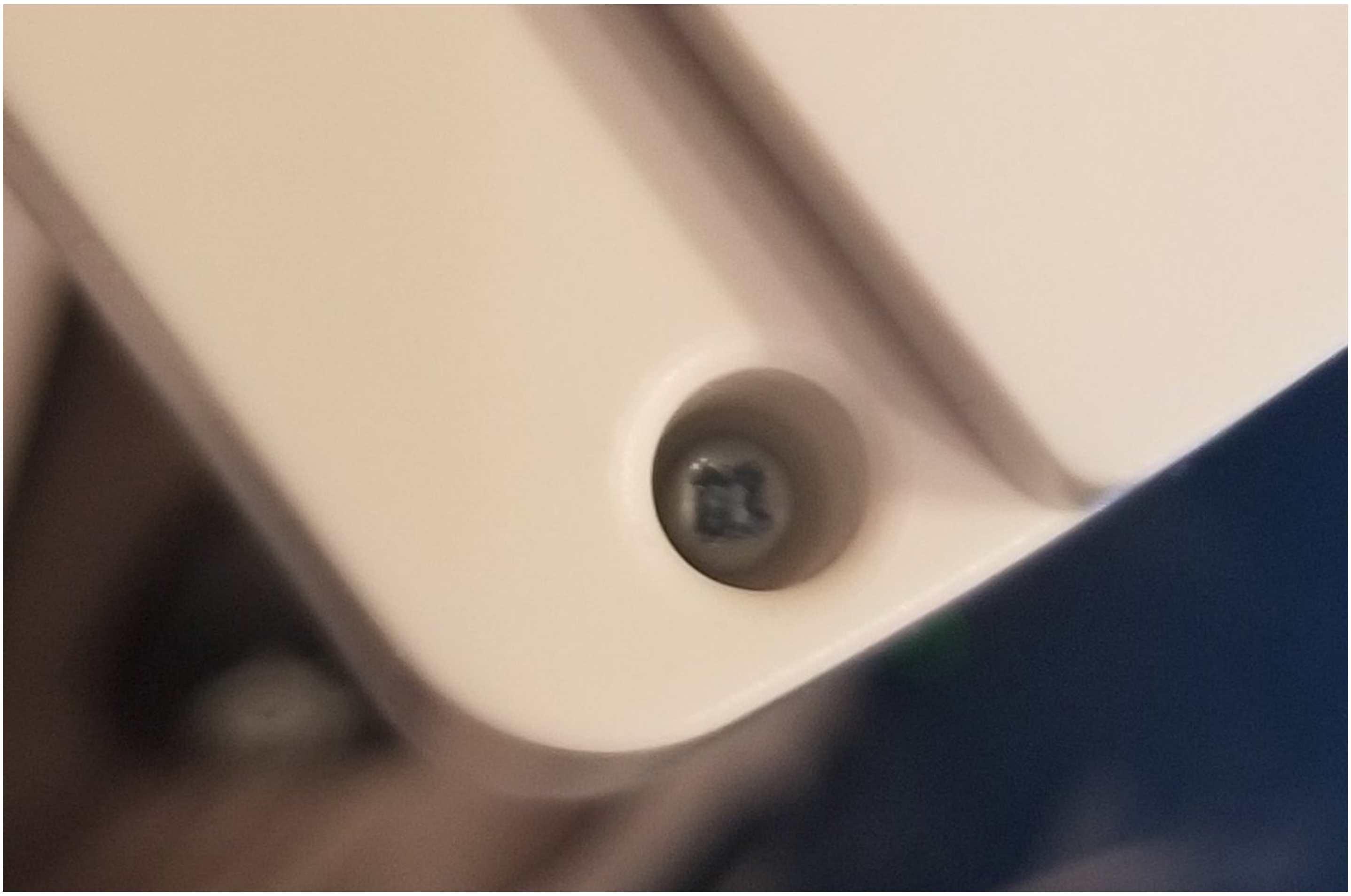


so the app wants to take pictures, access your location,
make calls, and access all your files.
and if you deny it, it just dumps you in the settings page to fix permissions, with no message.

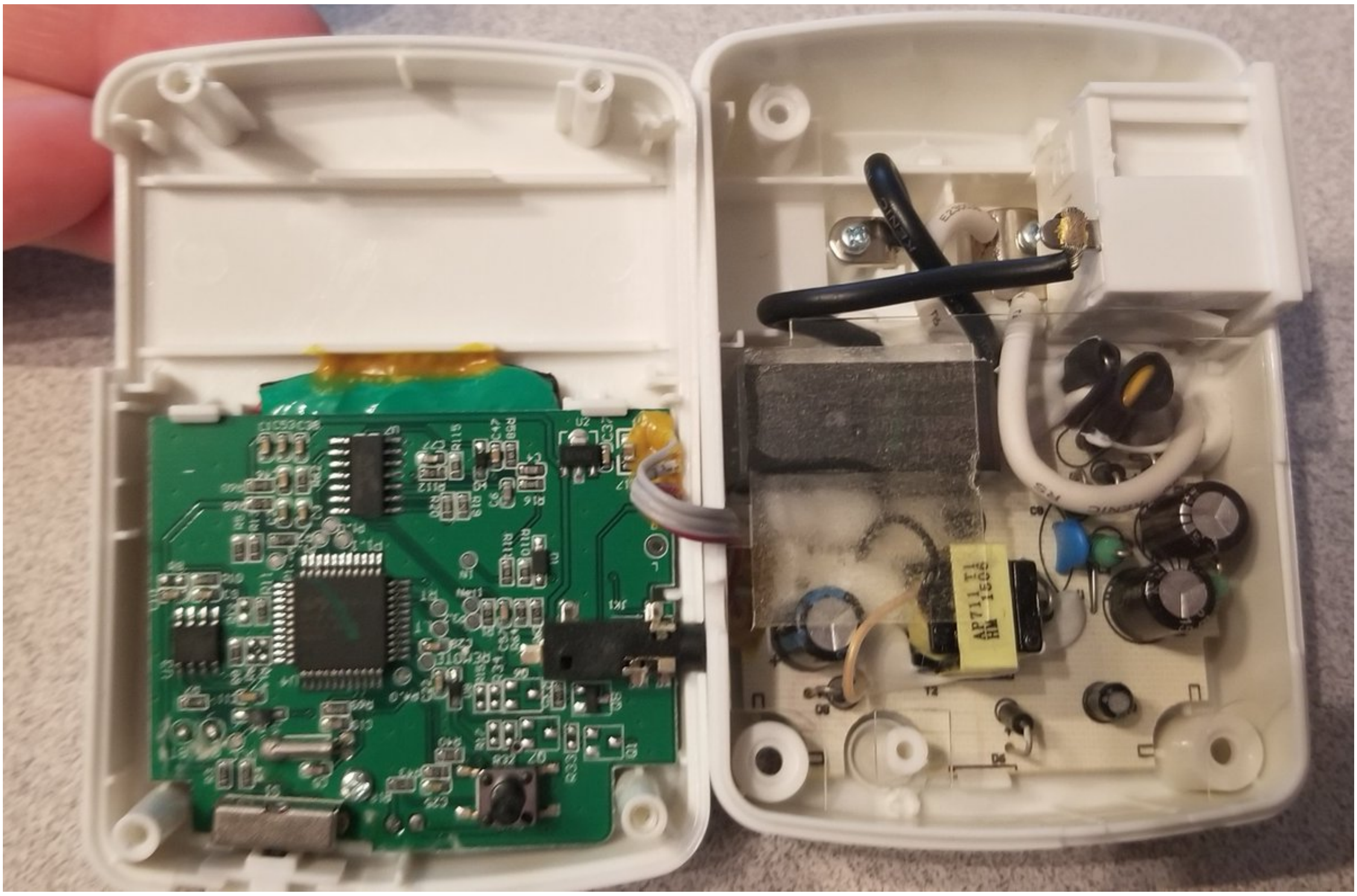


Allow **Timer** to take pictures and record video?

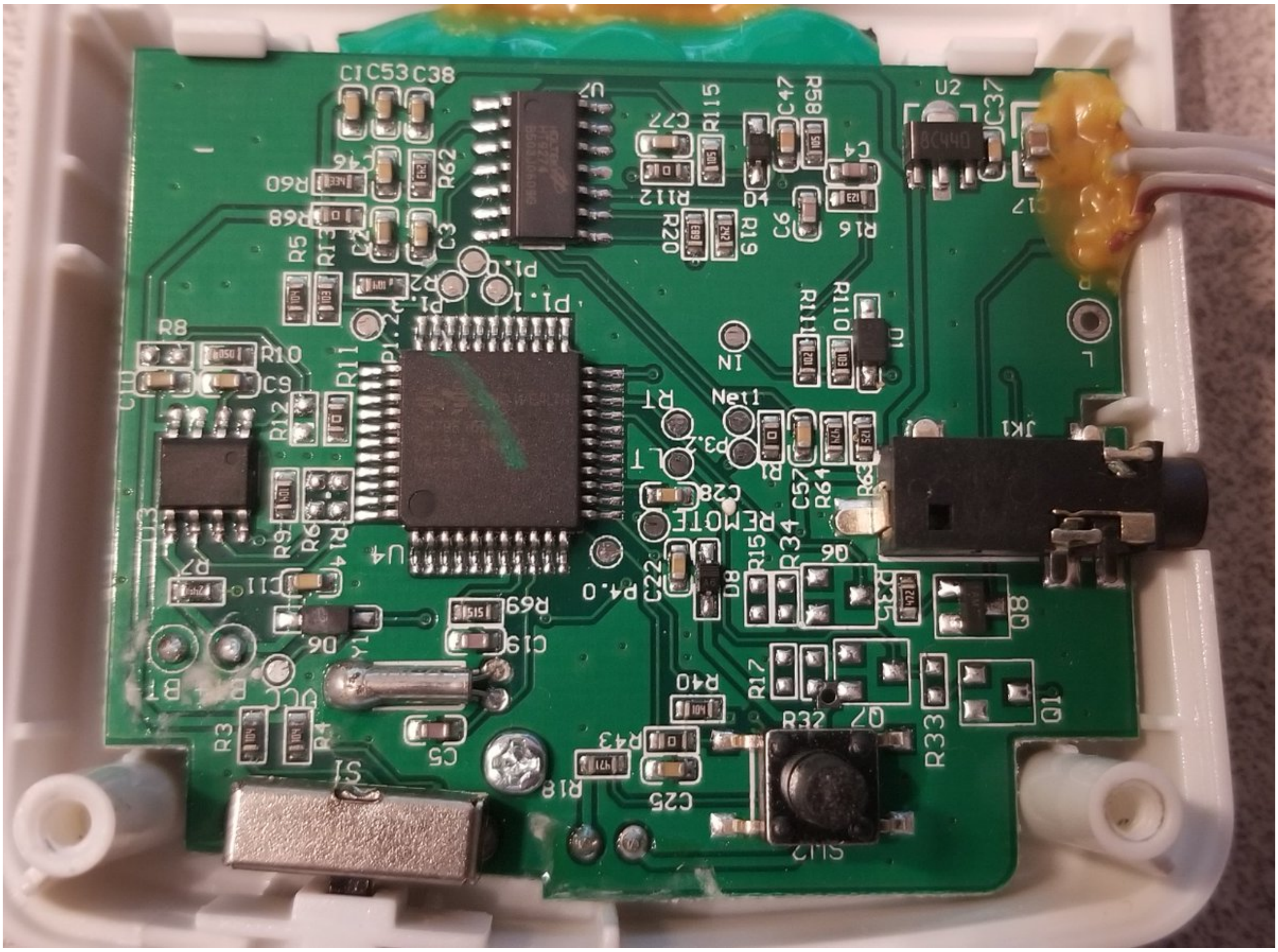
huh, not all of the screws are triangles.
there's 4 of them, and 2 are philips



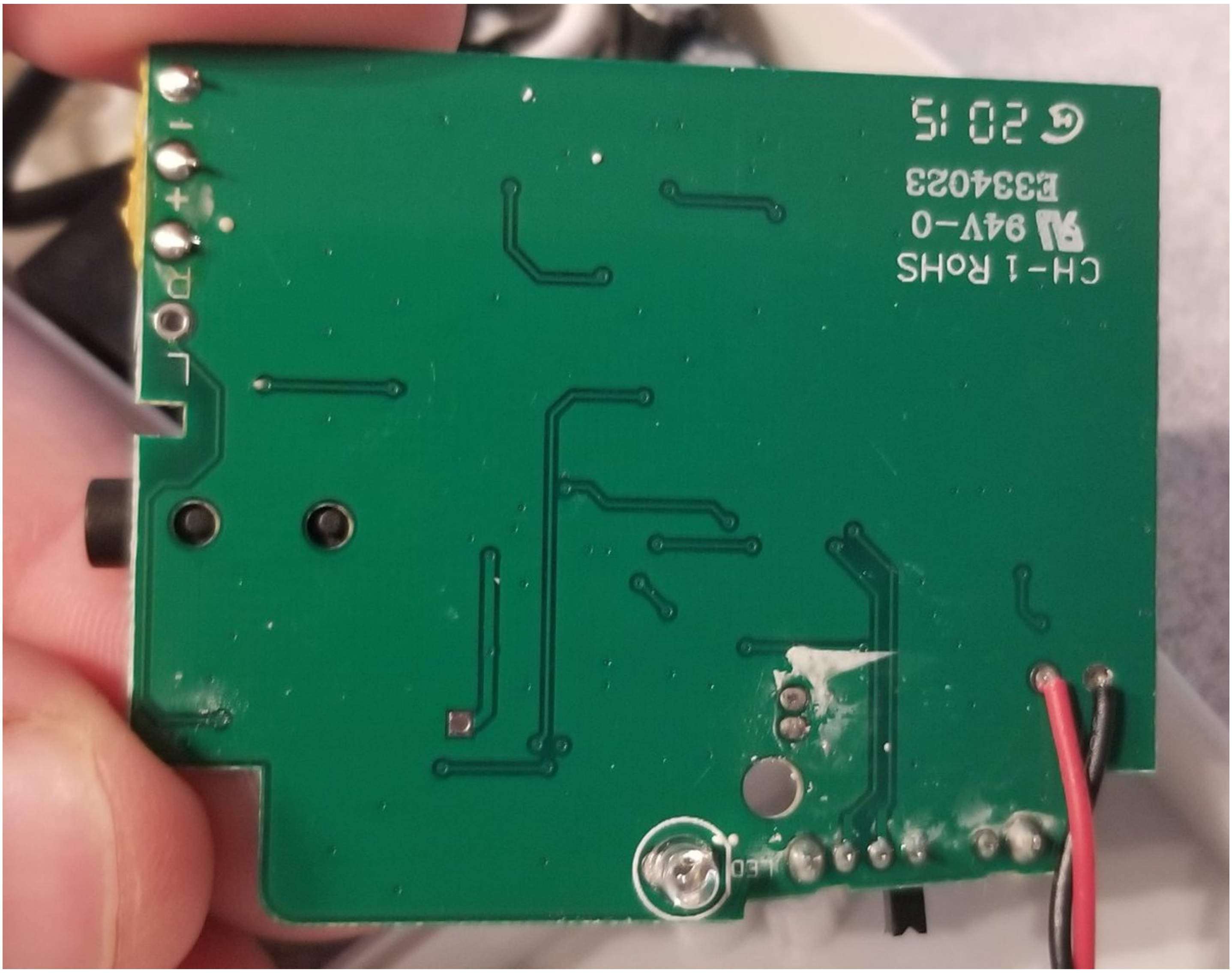
I hoped that'd make more sense when I opened it up, like one went into a PCB and the other didn't, but NOPE! it's just because Reasons.



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?

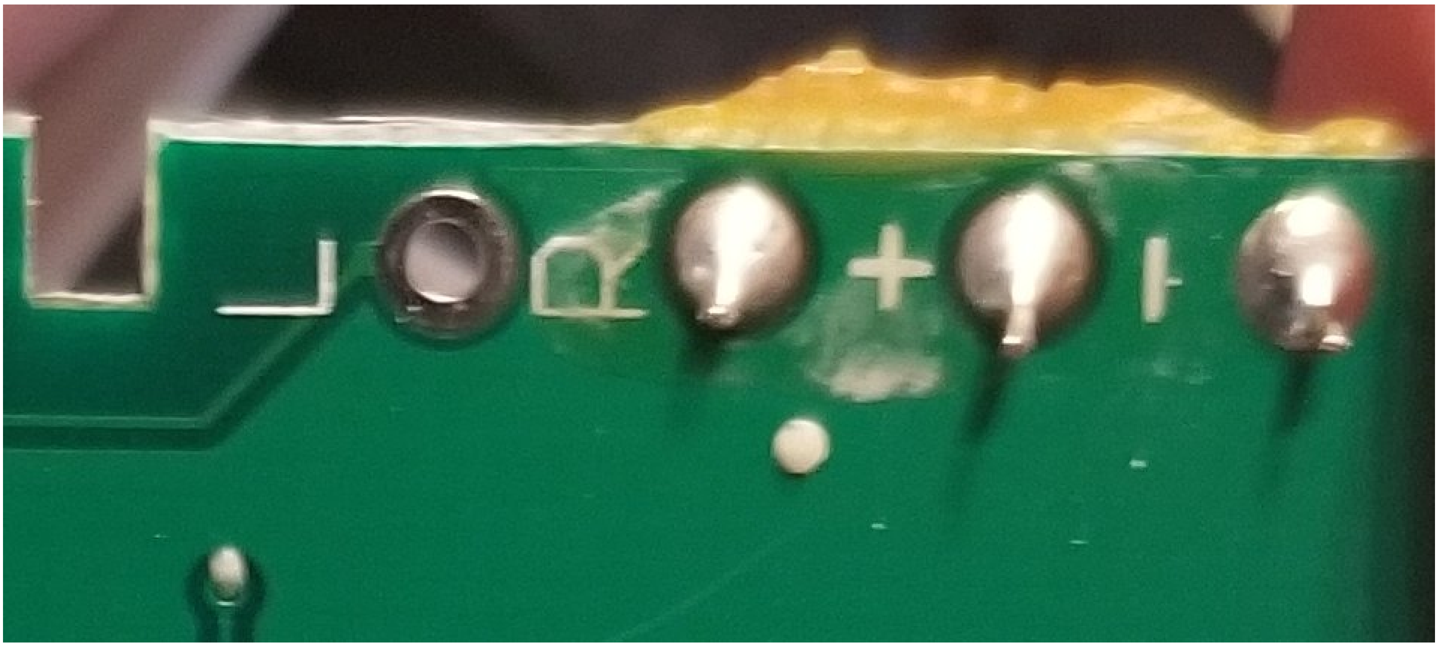


Nothing on the other side but the LED.

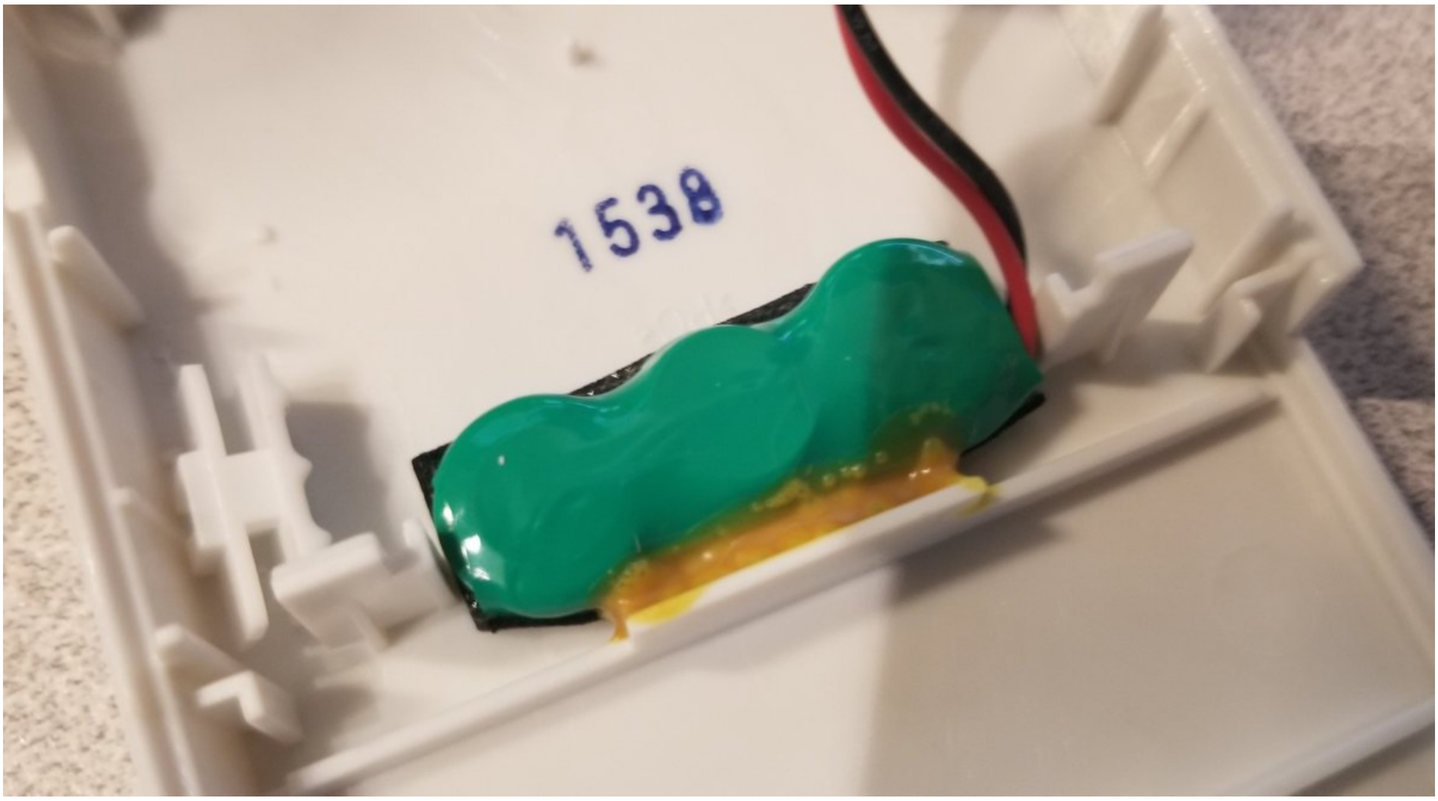


Although this bit is interesting: L/R/+/-, on the cables going to the other board.

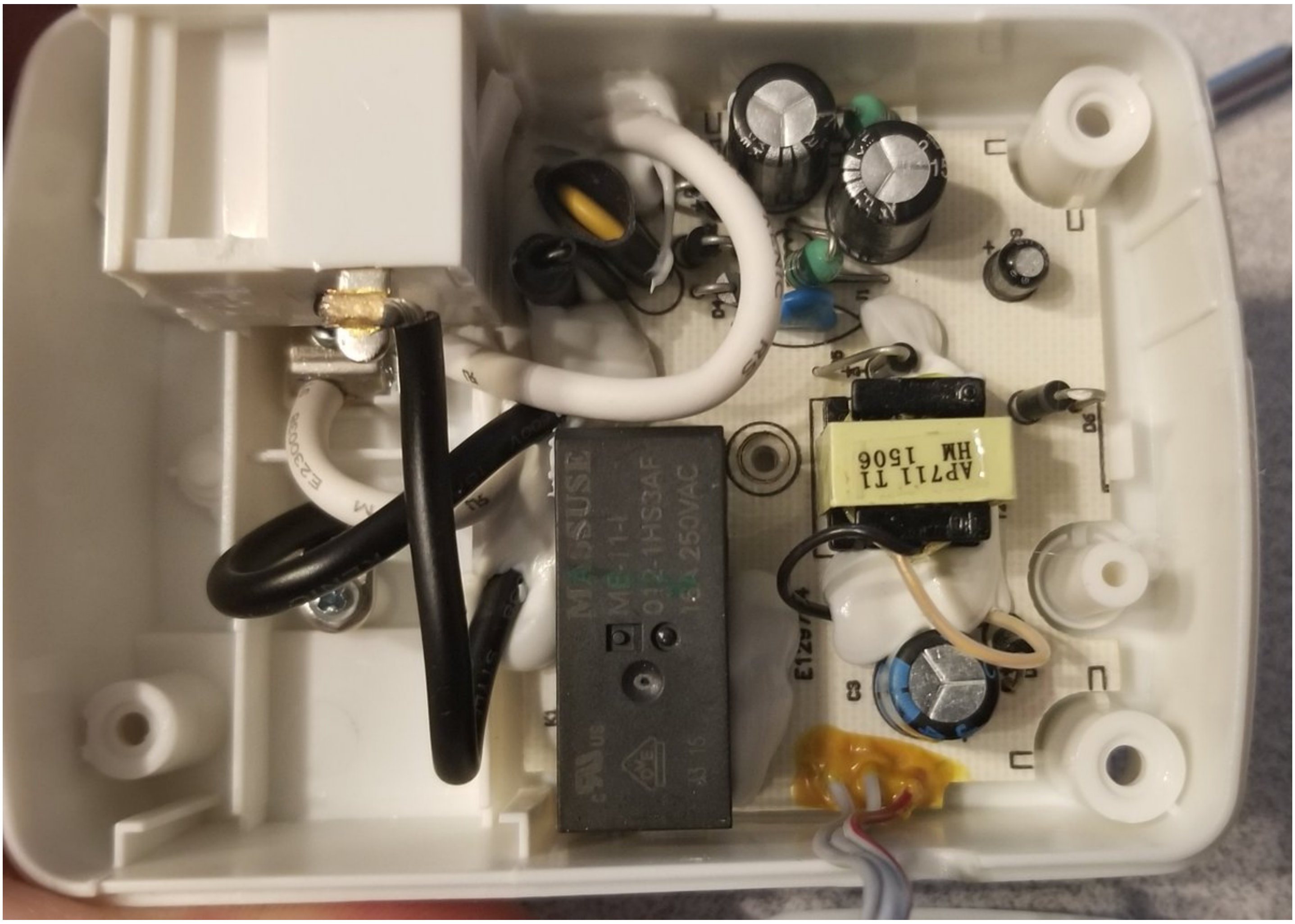
L isn't connected... I think that means there's a version of this that can control two outlets at once, not just one.



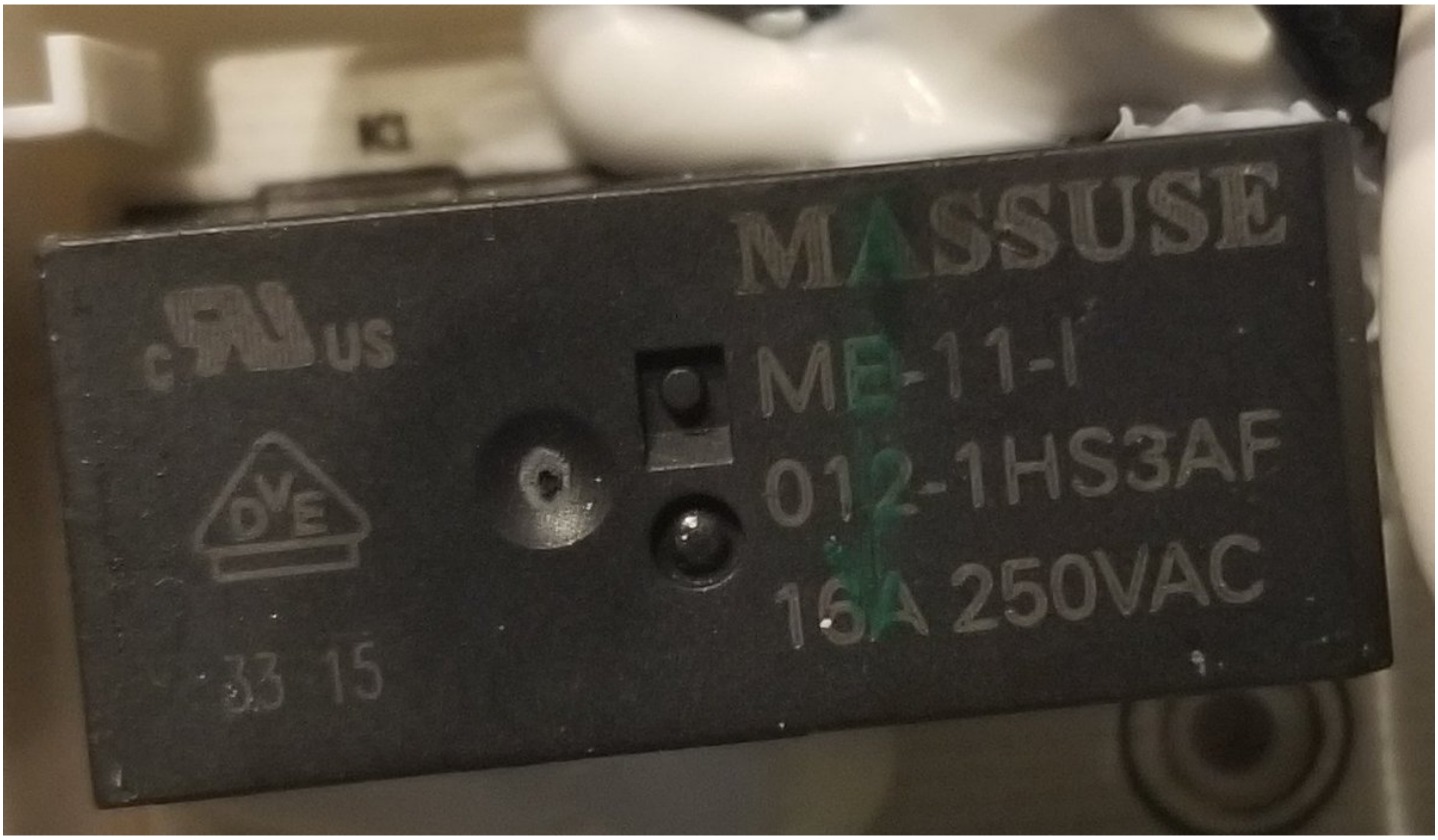
also HIDDEN BATTERY!
someday that will die and leak and the whole thing will be destroyed.



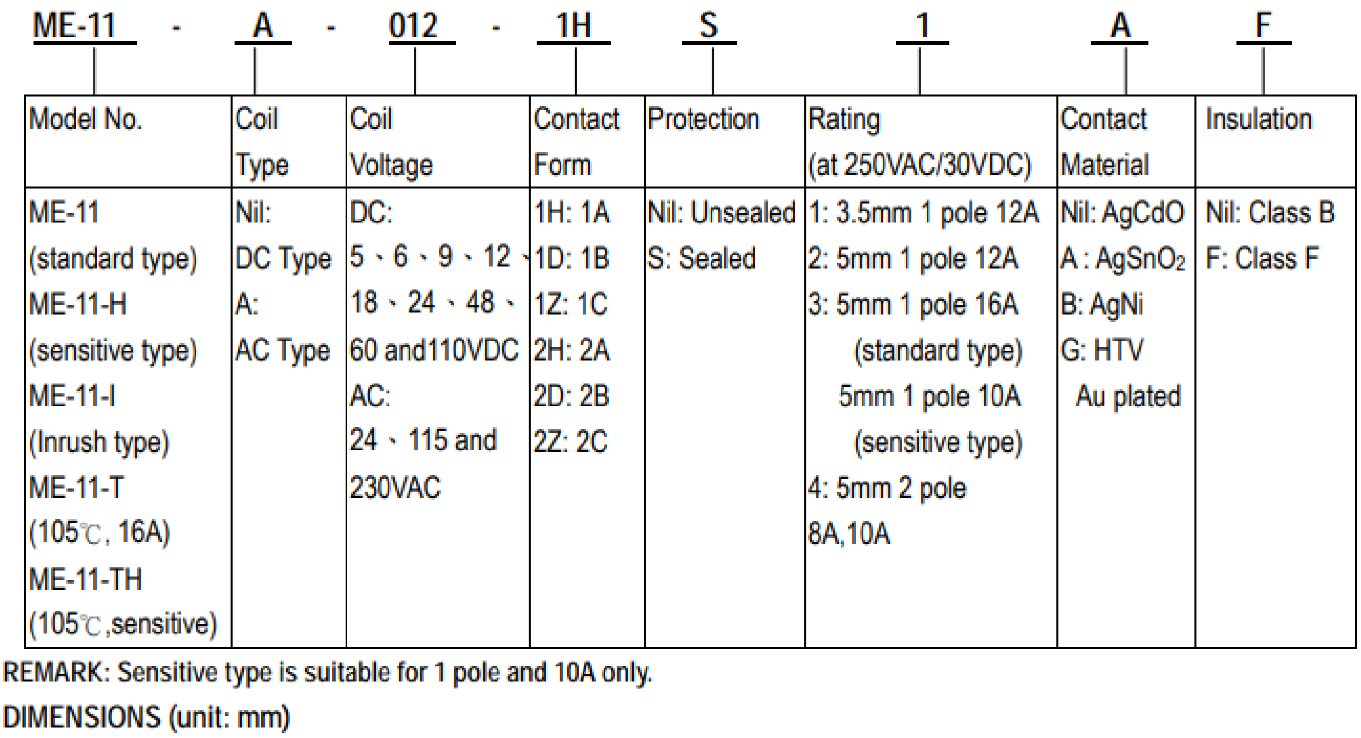
The other PCB.
I do like that they keep all the high-voltage AC stuff separate from the low-voltage DC stuff.
Cheaper versions of this would have just had one PCB.



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.



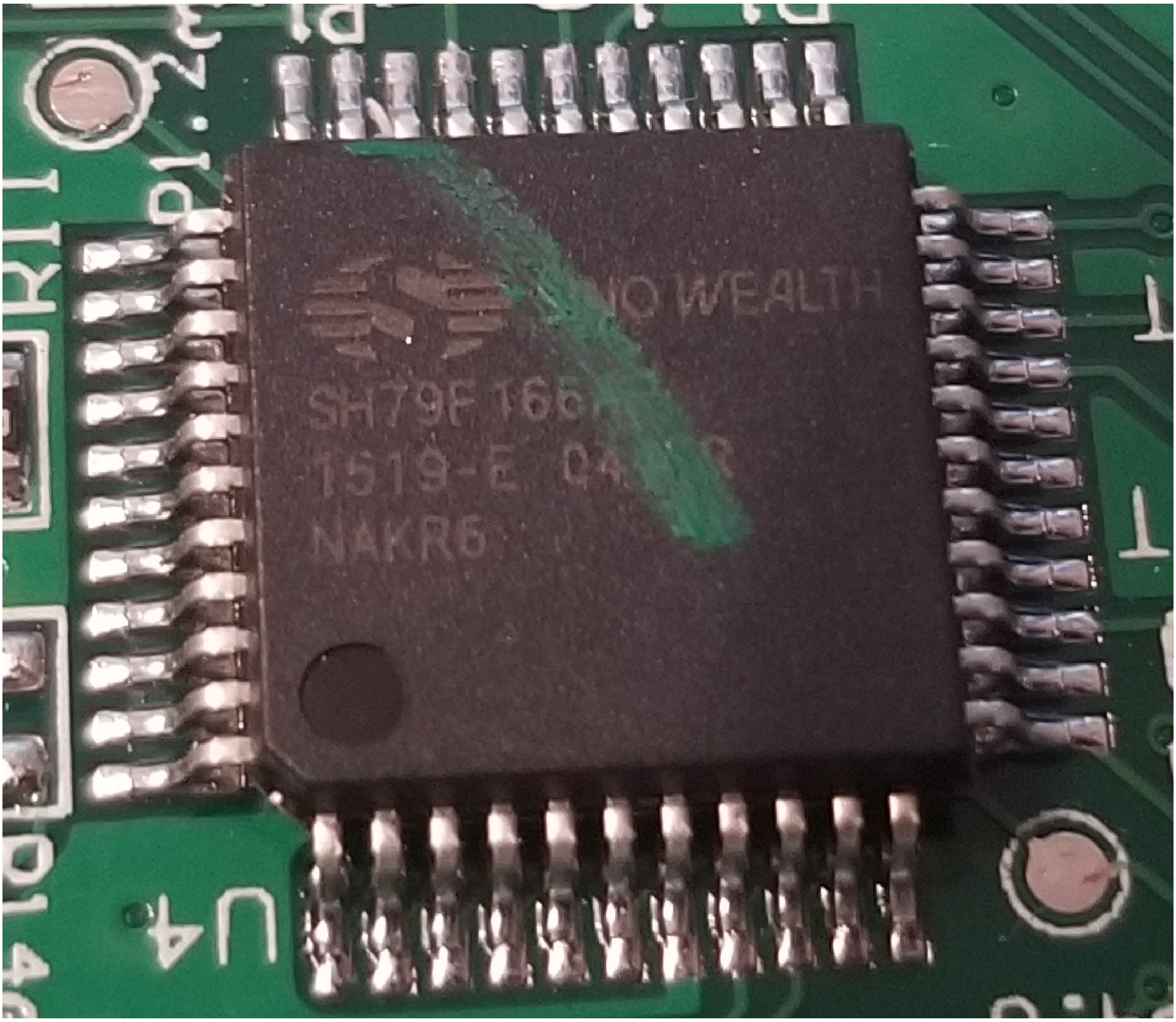
ME-11 - A - 012 - 1H S 1 A F

Model No.	Coil Type	Coil Voltage	Contact Form	Protection	Rating (at 250VAC/30VDC)	Contact Material	Insulation
ME-11 (standard type) ME-11-H (sensitive type) ME-11-I (Inrush type) ME-11-T (105℃, 16A) ME-11-TH (105℃,sensitive)	Nil: DC Type A: AC Type	DC: 5、6、9、12、18、24、48、60 and110VDC AC: 24、115 and 230VAC	1H: 1A 1D: 1B 1Z: 1C 2H: 2A 2D: 2B 2Z: 2C	Nil: Unsealed S: Sealed	1: 3.5mm 1 pole 12A 2: 5mm 1 pole 12A 3: 5mm 1 pole 16A (standard type) 5mm 1 pole 10A (sensitive type) 4: 5mm 2 pole 8A,10A	Nil: AgCdO A : AgSnO$_2$ B: AgNi G: HTV Au plated	Nil: Class B F: Class F

REMARK: Sensitive type is suitable for 1 pole and 10A only.

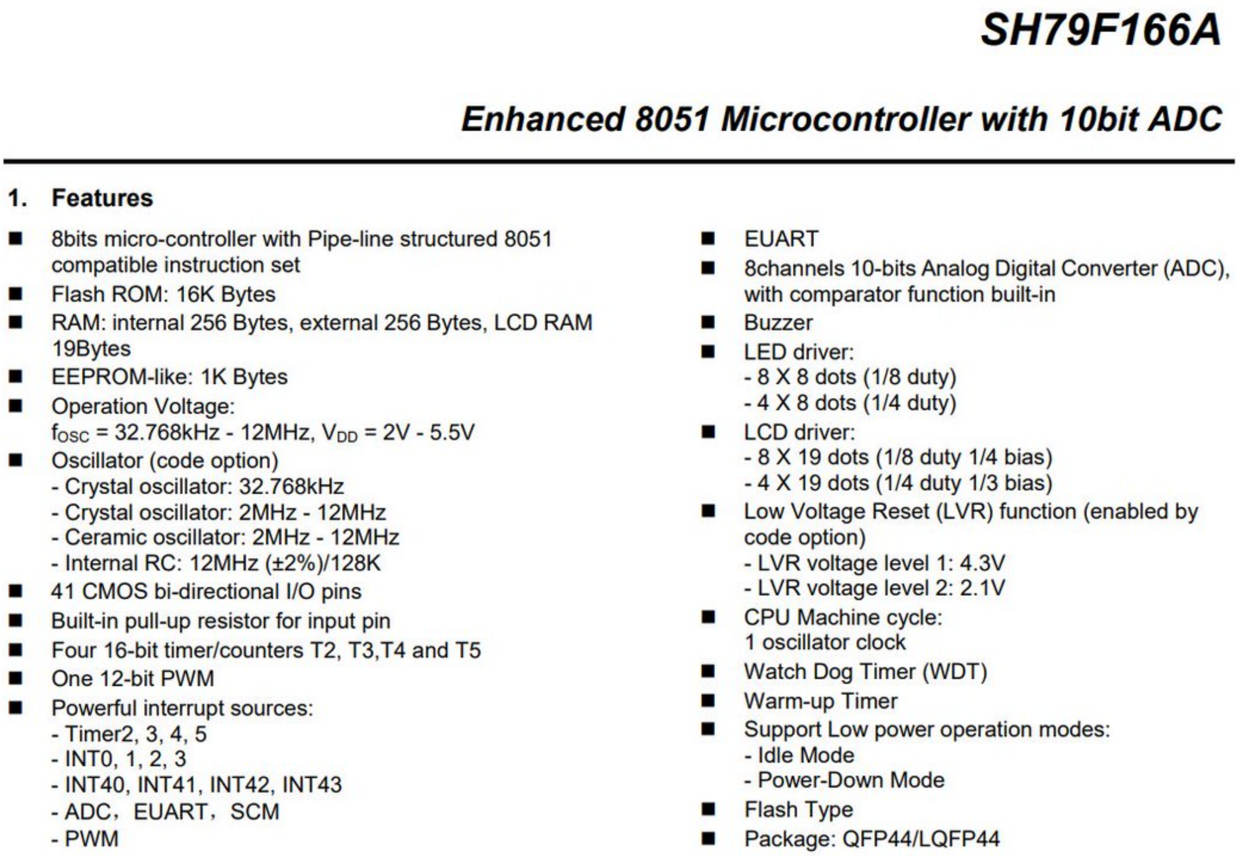
DIMENSIONS (unit: mm)

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



SH79F166A

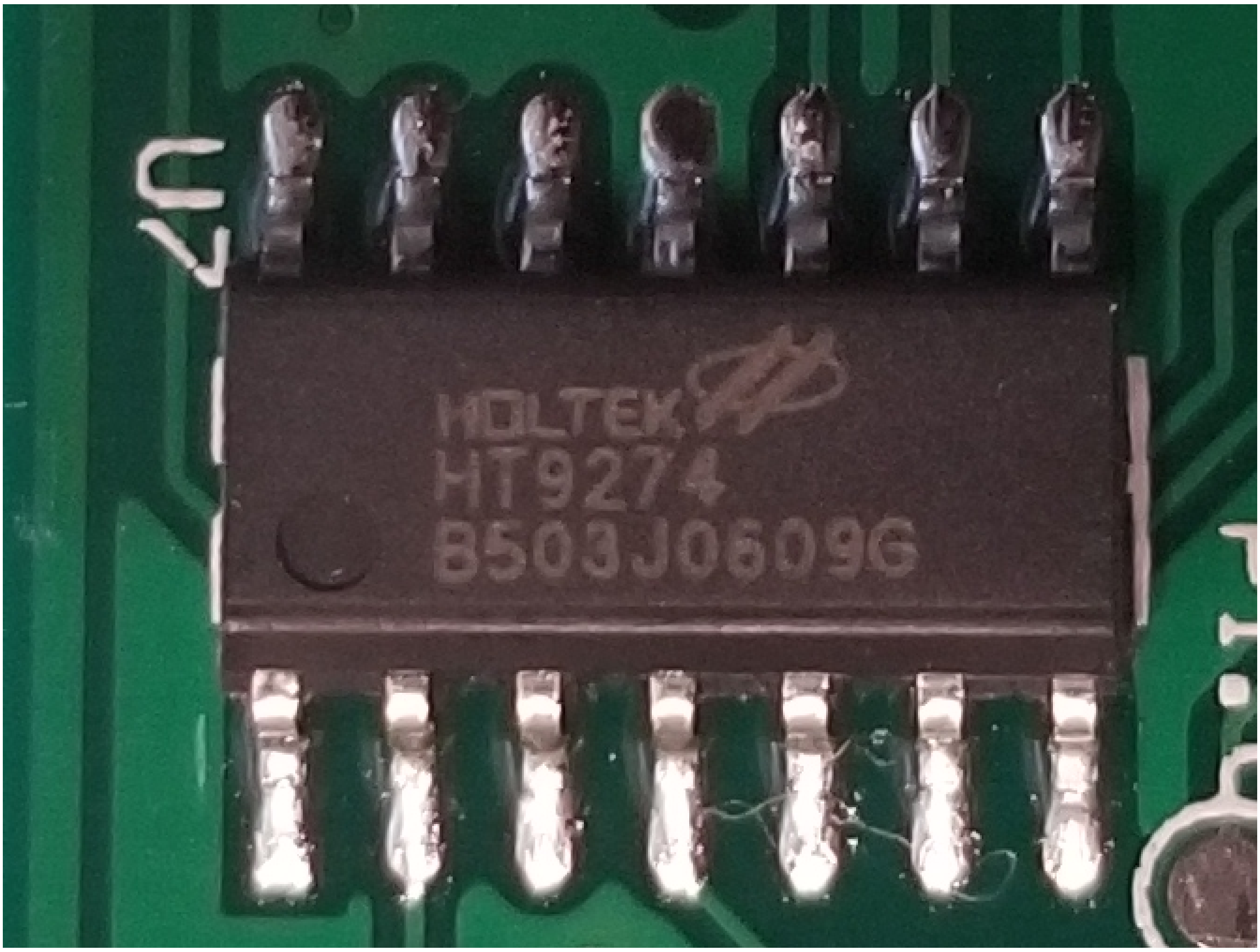
Enhanced 8051 Microcontroller with 10bit ADC

1. Features

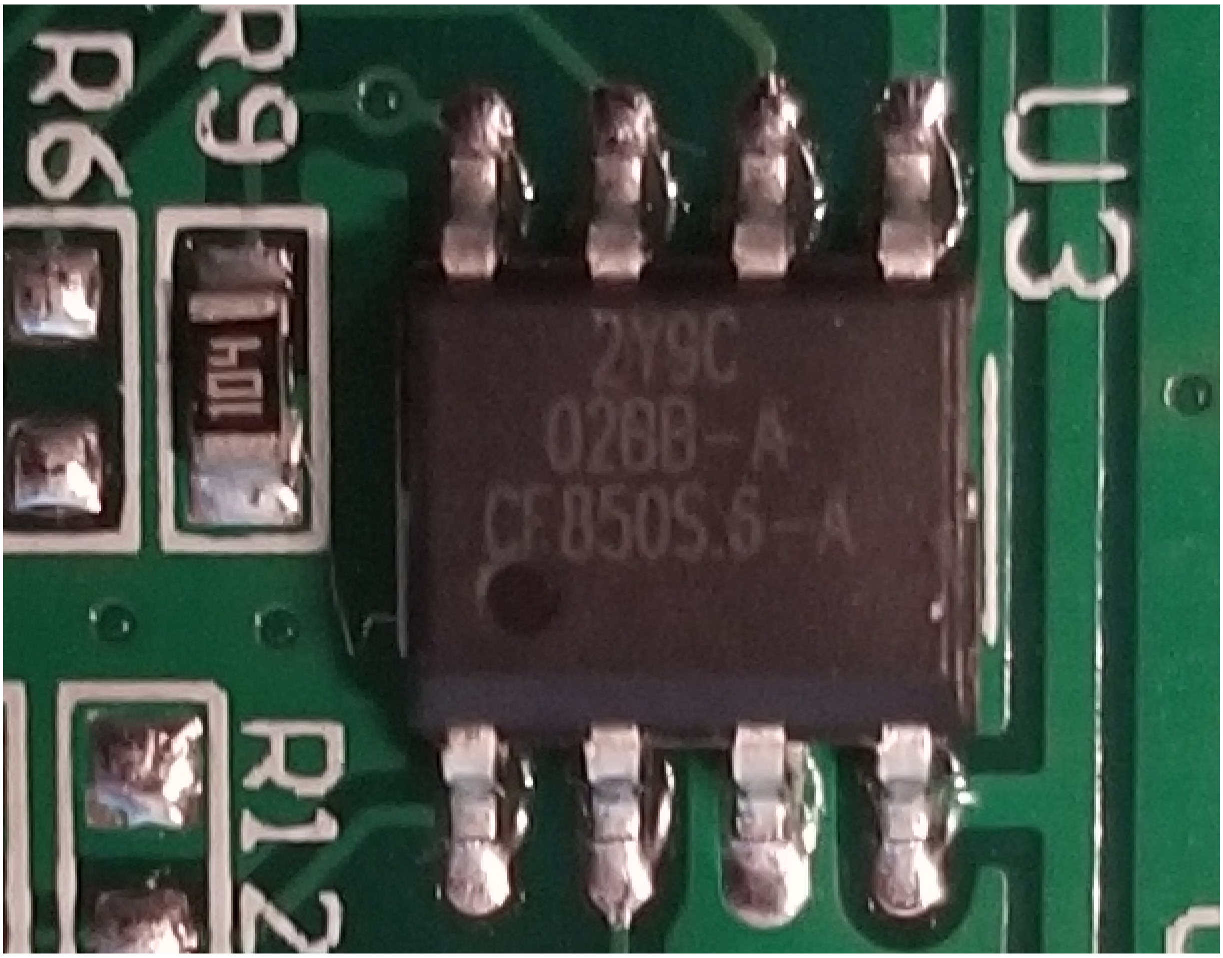
- 8bits micro-controller with Pipe-line structured 8051 compatible instruction set
- Flash ROM: 16K Bytes
- RAM: internal 256 Bytes, external 256 Bytes, LCD RAM 19Bytes
- EEPROM-like: 1K Bytes
- Operation Voltage:
  f_{OSC} = 32.768kHz - 12MHz, V_{DD} = 2V - 5.5V
- Oscillator (code option)
  - Crystal oscillator: 32.768kHz
  - Crystal oscillator: 2MHz - 12MHz
  - Ceramic oscillator: 2MHz - 12MHz
  - Internal RC: 12MHz (±2%)/128K
- 41 CMOS bi-directional I/O pins
- Built-in pull-up resistor for input pin
- Four 16-bit timer/counters T2, T3,T4 and T5
- One 12-bit PWM
- Powerful interrupt sources:
  - Timer2, 3, 4, 5
  - INT0, 1, 2, 3
  - INT40, INT41, INT42, INT43
  - ADC，EUART，SCM
  - PWM
- EUART
- 8channels 10-bits Analog Digital Converter (ADC), with comparator function built-in
- Buzzer
- LED driver:
  - 8 X 8 dots (1/8 duty)
  - 4 X 8 dots (1/4 duty)
- LCD driver:
  - 8 X 19 dots (1/8 duty 1/4 bias)
  - 4 X 19 dots (1/4 duty 1/3 bias)
- Low Voltage Reset (LVR) function (enabled by code option)
  - LVR voltage level 1: 4.3V
  - LVR voltage level 2: 2.1V
- CPU Machine cycle:
  1 oscillator clock
- Watch Dog Timer (WDT)
- Warm-up Timer
- Support Low power operation modes:
  - Idle Mode
  - Power-Down Mode
- Flash Type
- Package: QFP44/LQFP44

Over here is a Holtek HT9274.

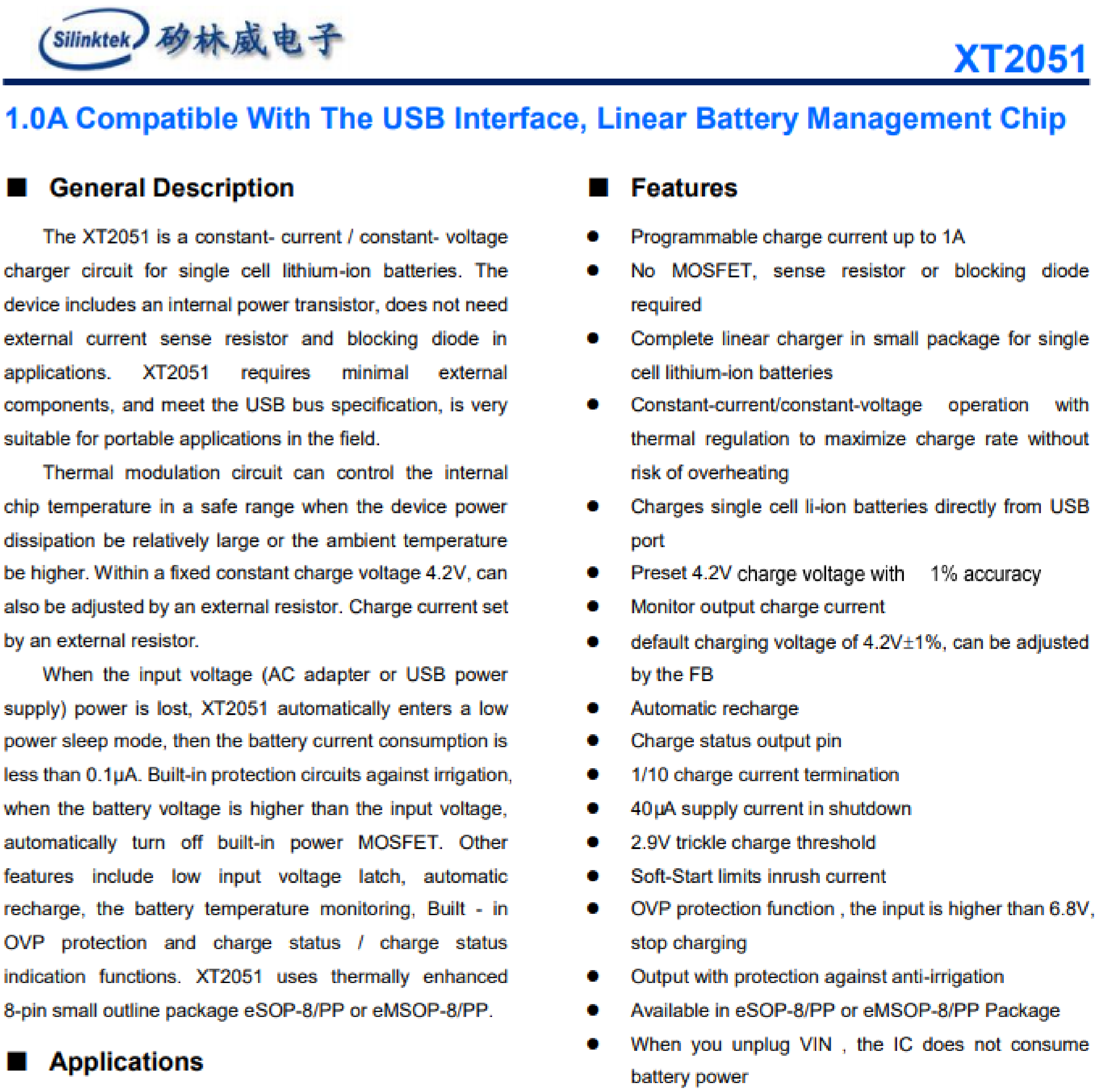
That's a quad-op-amp.



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.



Silinktek 矽林威电子

XT2051

1.0A Compatible With The USB Interface, Linear Battery Management Chip

■ General Description

The XT2051 is a constant- current / constant- voltage charger circuit for single cell lithium-ion batteries. The device includes an internal power transistor, does not need external current sense resistor and blocking diode in applications. XT2051 requires minimal external components, and meet the USB bus specification, is very suitable for portable applications in the field.

Thermal modulation circuit can control the internal chip temperature in a safe range when the device power dissipation be relatively large or the ambient temperature be higher. Within a fixed constant charge voltage 4.2V, can also be adjusted by an external resistor. Charge current set by an external resistor.

When the input voltage (AC adapter or USB power supply) power is lost, XT2051 automatically enters a low power sleep mode, then the battery current consumption is less than 0.1μA. Built-in protection circuits against irrigation, when the battery voltage is higher than the input voltage, automatically turn off built-in power MOSFET. Other features include low input voltage latch, automatic recharge, the battery temperature monitoring, Built - in OVP protection and charge status / charge status indication functions. XT2051 uses thermally enhanced 8-pin small outline package eSOP-8/PP or eMSOP-8/PP.

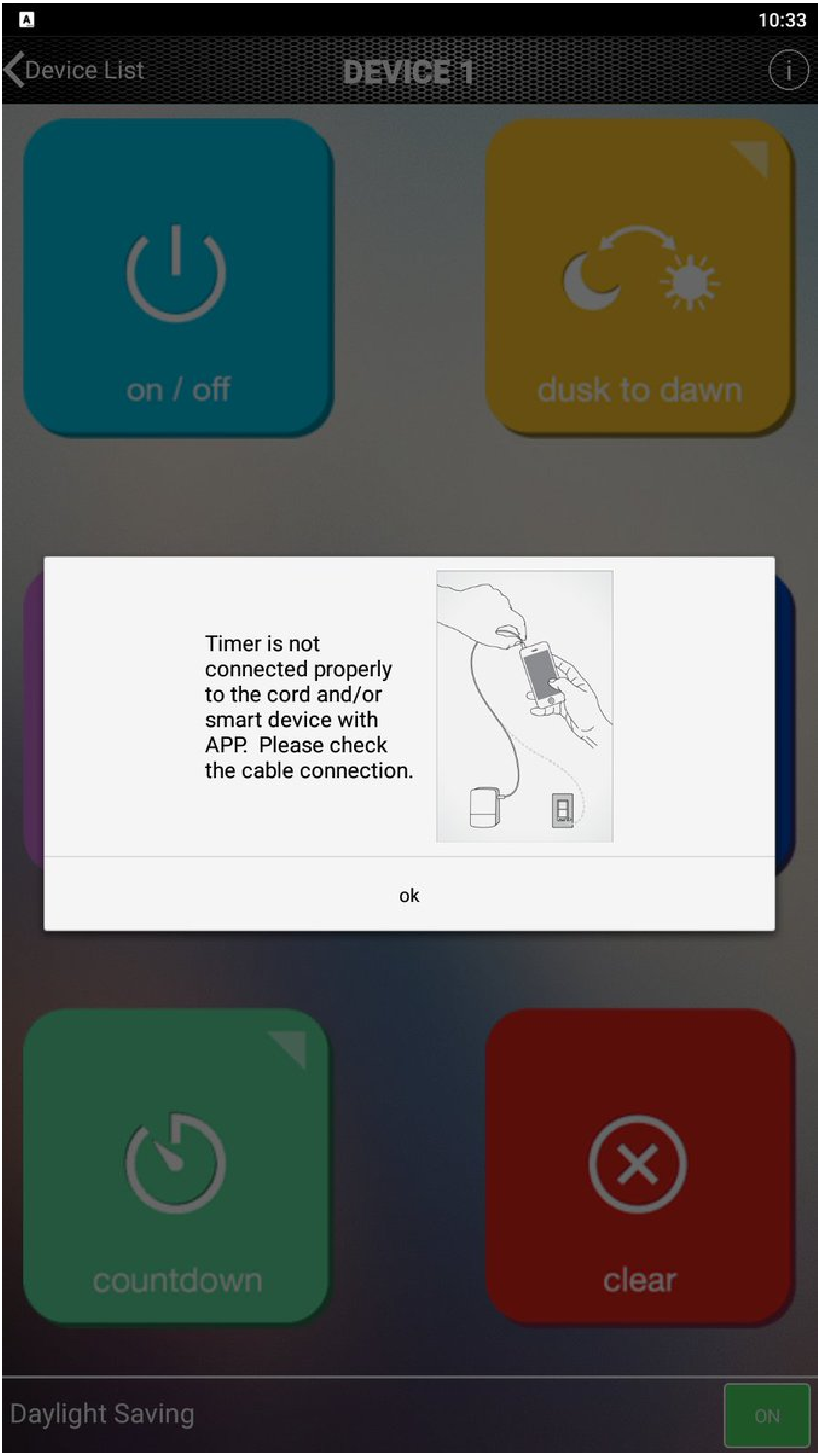
■ Applications

■ Features

- Programmable charge current up to 1A
- No MOSFET, sense resistor or blocking diode required
- Complete linear charger in small package for single cell lithium-ion batteries
- Constant-current/constant-voltage operation with thermal regulation to maximize charge rate without risk of overheating
- Charges single cell li-ion batteries directly from USB port
- Preset 4.2V charge voltage with 1% accuracy
- Monitor output charge current
- default charging voltage of 4.2V±1%, can be adjusted by the FB
- Automatic recharge
- Charge status output pin
- 1/10 charge current termination
- 40μA supply current in shutdown
- 2.9V trickle charge threshold
- Soft-Start limits inrush current
- OVP protection function , the input is higher than 6.8V, stop charging
- Output with protection against anti-irrigation
- Available in eSOP-8/PP or eMSOP-8/PP Package
- When you unplug VIN , the IC does not consume battery power

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

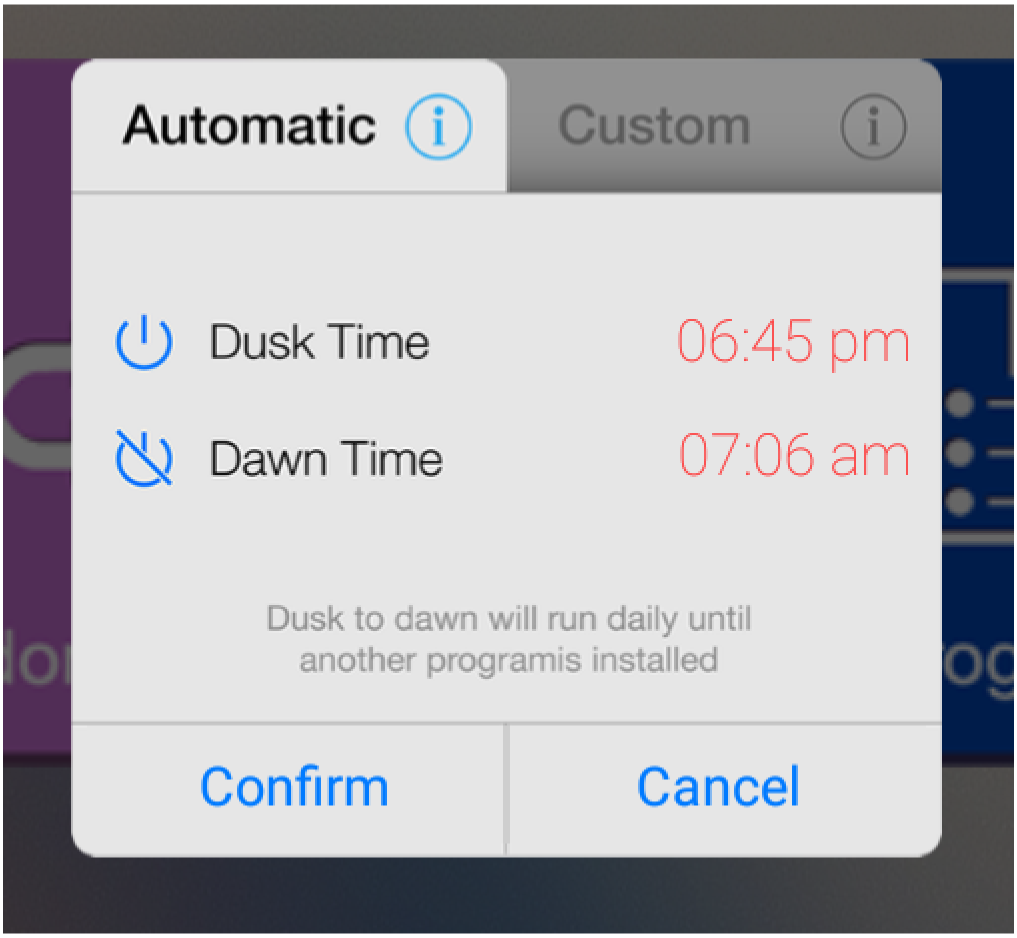
so apparently the communication with the phone/tablet is two way!
because it can tell it's not connected properly, in this emulator I'm using



Timer is not connected properly to the cord and/or smart device with APP. Please check the cable connection.

Daylight Saving ON

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

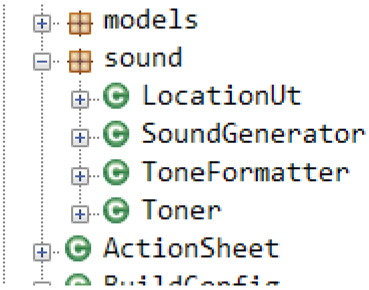


Dusk Time 06:45 pm

Dawn Time 07:06 am

Dusk to dawn will run daily until
another programis installed

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.

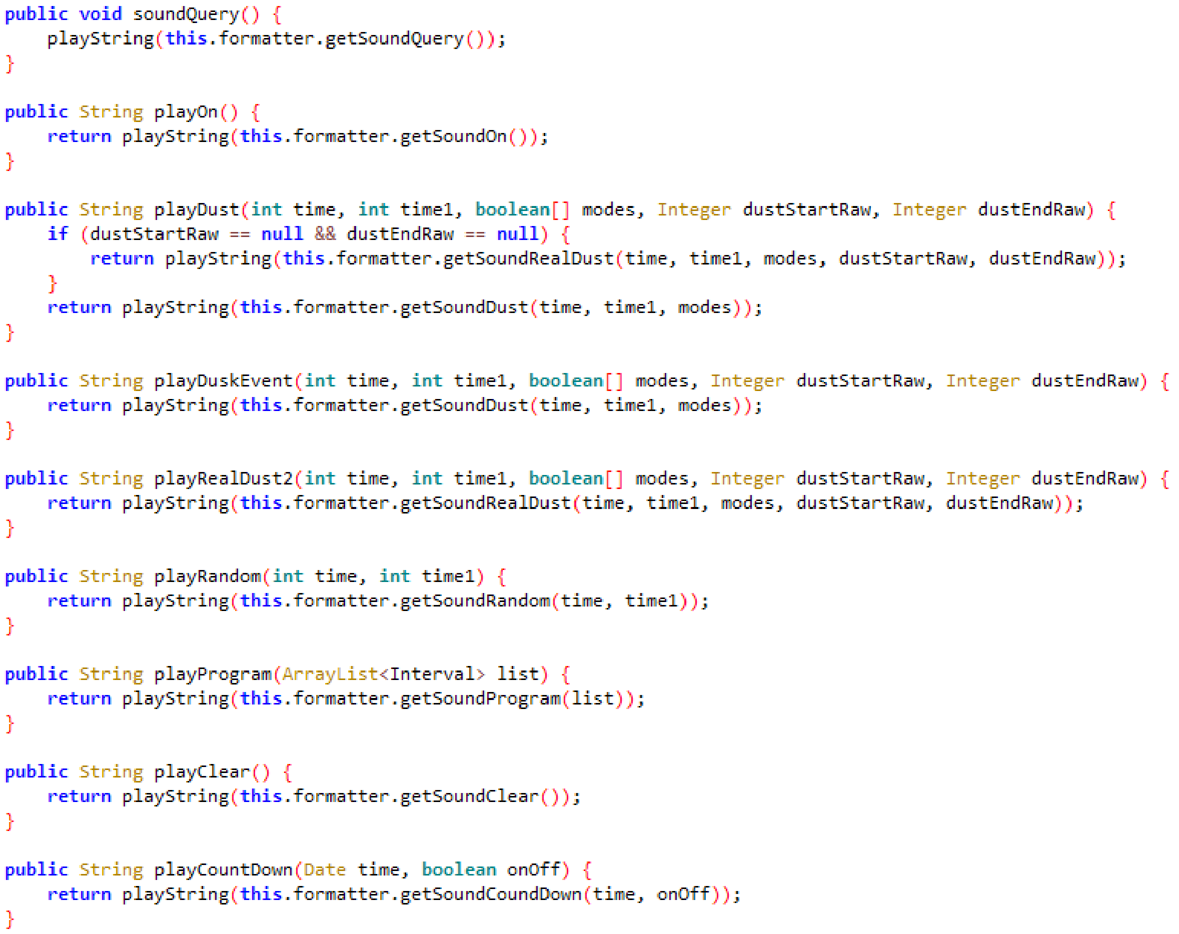


```
models
sound
    LocationUt
    SoundGenerator
    ToneFormatter
    Toner
ActionSheet
BuildConfig
```

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.

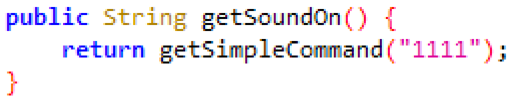
sadly since it's gaming oriented it doesn't seem to have any way to shim that out.

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)



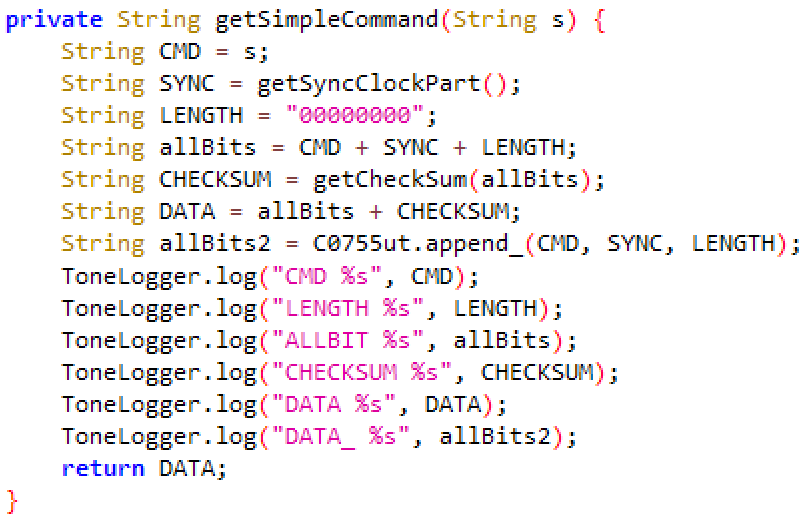
```
public void soundQuery() {
    playString(this.formatter.getSoundQuery());
}

public String playOn() {
    return playString(this.formatter.getSoundOn());
}

public String playDust(int time, int time1, boolean[] modes, Integer dustStartRaw, Integer dustEndRaw) {
    if (dustStartRaw == null && dustEndRaw == null) {
        return playString(this.formatter.getSoundRealDust(time, time1, modes, dustStartRaw, dustEndRaw));
    }
    return playString(this.formatter.getSoundDust(time, time1, modes));
}

public String playDuskEvent(int time, int time1, boolean[] modes, Integer dustStartRaw, Integer dustEndRaw) {
    return playString(this.formatter.getSoundDust(time, time1, modes));
}

public String playRealDust2(int time, int time1, boolean[] modes, Integer dustStartRaw, Integer dustEndRaw) {
    return playString(this.formatter.getSoundRealDust(time, time1, modes, dustStartRaw, dustEndRaw));
}

public String playRandom(int time, int time1) {
    return playString(this.formatter.getSoundRandom(time, time1));
}

public String playProgram(ArrayList<Interval> list) {
    return playString(this.formatter.getSoundProgram(list));
}

public String playClear() {
    return playString(this.formatter.getSoundClear());
}

public String playCountDown(Date time, boolean onOff) {
    return playString(this.formatter.getSoundCoundDown(time, onOff));
}
```

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



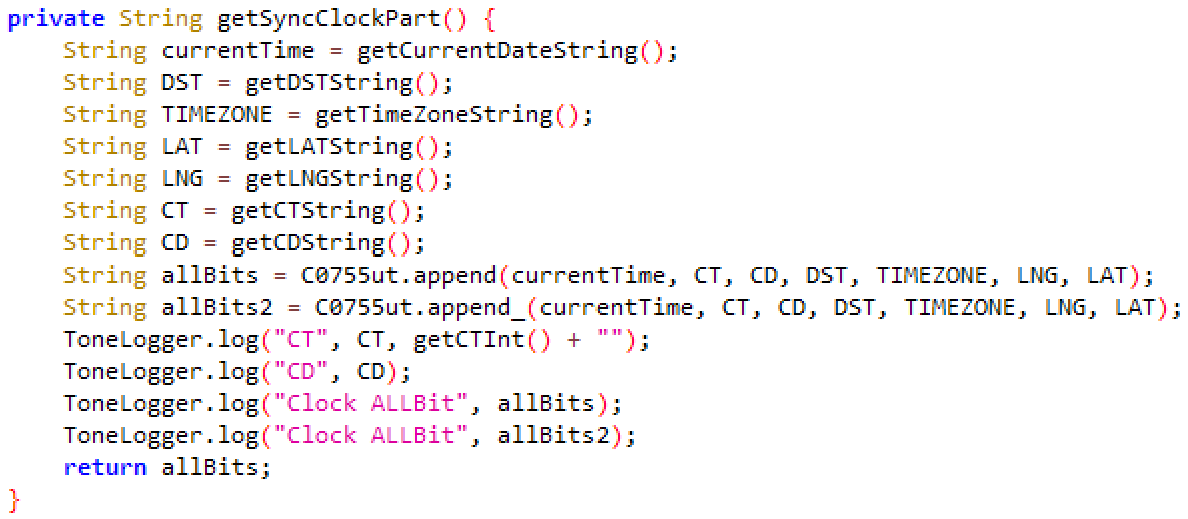
```
public String getSoundOn() {
    return getSimpleCommand("1111");
}
```

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.



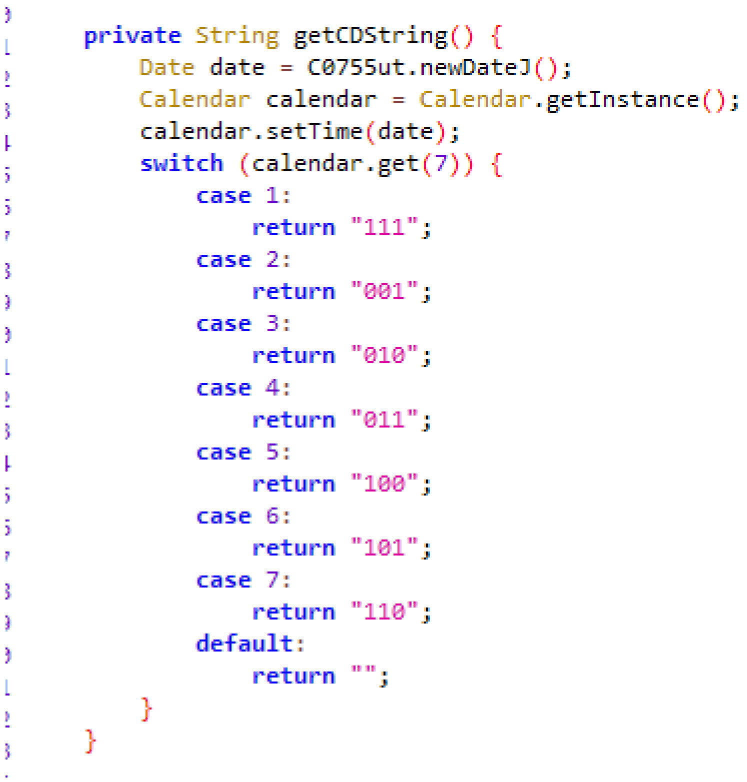
```
private String getSimpleCommand(String s) {
    String CMD = s;
    String SYNC = getSyncClockPart();
    String LENGTH = "00000000";
    String allBits = CMD + SYNC + LENGTH;
    String CHECKSUM = getCheckSum(allBits);
    String DATA = allBits + CHECKSUM;
    String allBits2 = C0755ut.append_(CMD, SYNC, LENGTH);
    ToneLogger.log("CMD %s", CMD);
    ToneLogger.log("LENGTH %s", LENGTH);
    ToneLogger.log("ALLBIT %s", allBits);
    ToneLogger.log("CHECKSUM %s", CHECKSUM);
    ToneLogger.log("DATA %s", DATA);
    ToneLogger.log("DATA_ %s", allBits2);
    return DATA;
}
```

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.



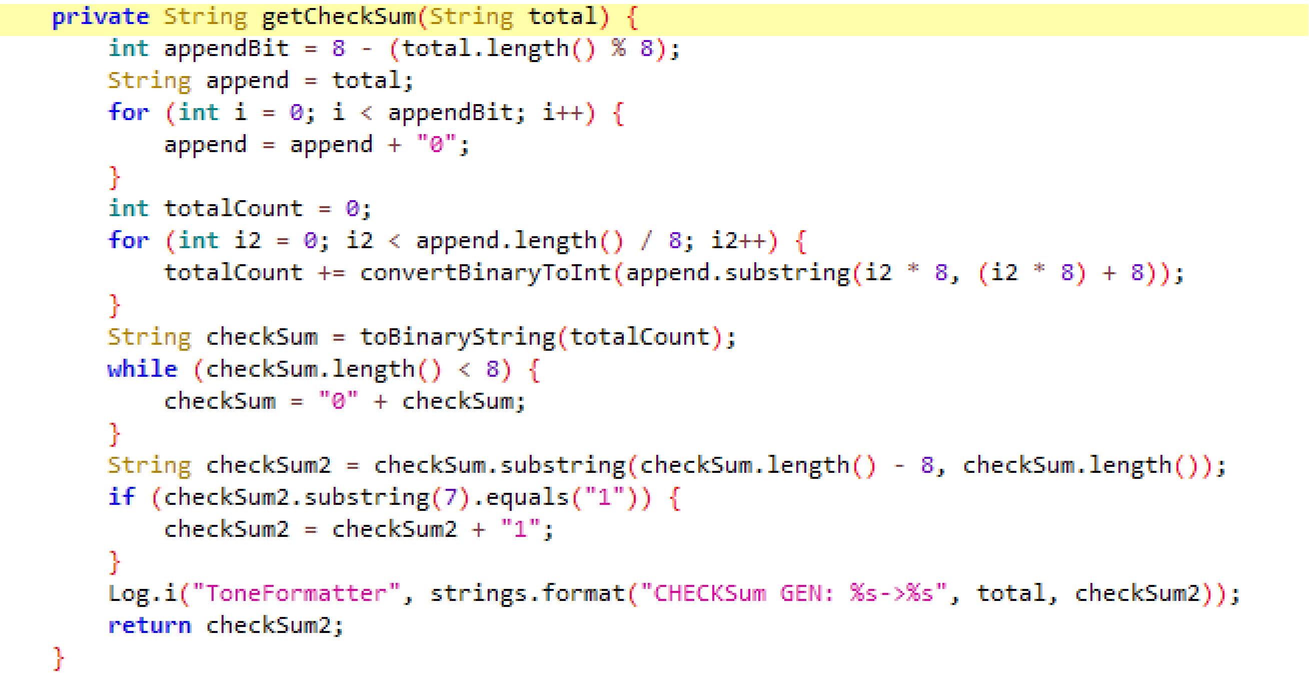
```java
private String getSyncClockPart() {
    String currentTime = getCurrentDateString();
    String DST = getDSTString();
    String TIMEZONE = getTimeZoneString();
    String LAT = getLATString();
    String LNG = getLNGString();
    String CT = getCTString();
    String CD = getCDString();
    String allBits = C0755ut.append(currentTime, CT, CD, DST, TIMEZONE, LNG, LAT);
    String allBits2 = C0755ut.append_(currentTime, CT, CD, DST, TIMEZONE, LNG, LAT);
    ToneLogger.log("CT", CT, getCTInt() + "");
    ToneLogger.log("CD", CD);
    ToneLogger.log("Clock ALLBit", allBits);
    ToneLogger.log("Clock ALLBit", allBits2);
    return allBits;
}
```

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)



```java
private String getCDString() {
    Date date = C0755ut.newDateJ();
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(date);
    switch (calendar.get(7)) {
        case 1:
            return "111";
        case 2:
            return "001";
        case 3:
            return "010";
        case 4:
            return "011";
        case 5:
            return "100";
        case 6:
            return "101";
        case 7:
            return "110";
        default:
            return "";
    }
}
```

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.



```
    private String getCheckSum(String total) {
        int appendBit = 8 - (total.length() % 8);
        String append = total;
        for (int i = 0; i < appendBit; i++) {
            append = append + "0";
        }
        int totalCount = 0;
        for (int i2 = 0; i2 < append.length() / 8; i2++) {
            totalCount += convertBinaryToInt(append.substring(i2 * 8, (i2 * 8) + 8));
        }
        String checkSum = toBinaryString(totalCount);
        while (checkSum.length() < 8) {
            checkSum = "0" + checkSum;
        }
        String checkSum2 = checkSum.substring(checkSum.length() - 8, checkSum.length());
        if (checkSum2.substring(7).equals("1")) {
            checkSum2 = checkSum2 + "1";
        }
        Log.i("ToneFormatter", strings.format("CHECKSum GEN: %s->%s", total, checkSum2));
        return checkSum2;
    }
```

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

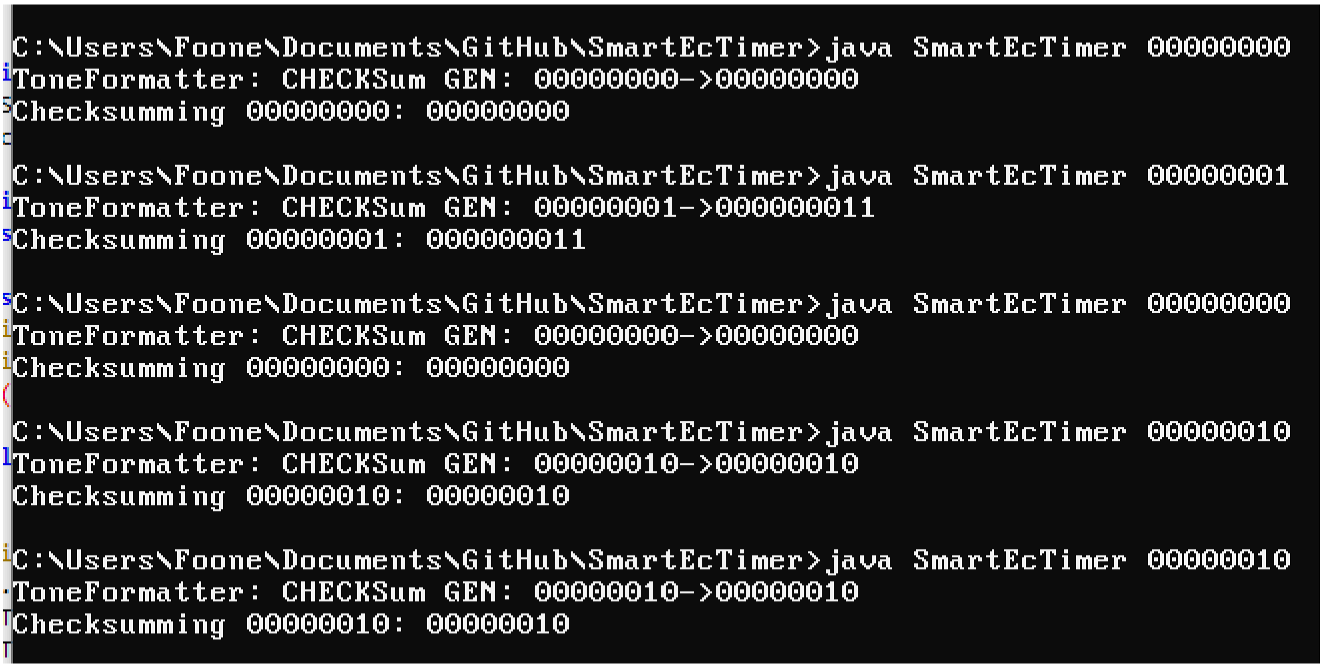
then it chops the checksum down to 8 bits... and checks if the last digit is a 1.

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.



```
C:\Users\Foone\Documents\GitHub\SmartEcTimer>java SmartEcTimer 00000000
ToneFormatter: CHECKSum GEN: 00000000->00000000
Checksumming 00000000: 00000000

C:\Users\Foone\Documents\GitHub\SmartEcTimer>java SmartEcTimer 00000001
ToneFormatter: CHECKSum GEN: 00000001->000000011
Checksumming 00000001: 000000011

C:\Users\Foone\Documents\GitHub\SmartEcTimer>java SmartEcTimer 00000000
ToneFormatter: CHECKSum GEN: 00000000->00000000
Checksumming 00000000: 00000000

C:\Users\Foone\Documents\GitHub\SmartEcTimer>java SmartEcTimer 00000010
ToneFormatter: CHECKSum GEN: 00000010->00000010
Checksumming 00000010: 00000010

C:\Users\Foone\Documents\GitHub\SmartEcTimer>java SmartEcTimer 00000010
ToneFormatter: CHECKSum GEN: 00000010->00000010
Checksumming 00000010: 00000010
```