Designing with Kicad

This is not a Tutorial

The web is full of tuorials: just pick one

- If unhappy pick another

Or you can run the application and read help

- As a general rule, I tend to prefer real documentation
- If documentation accompanies the program, then it is the best one

This is just a collection of ideas and suggestions

- You won't became an expert kicad user by reading this
- Moreover, you may as well disagree with me

Overview of Kicad

Kicad is a Free Sofware application to design PCBs

- It covers both schematic design and PCB drawing
- It has some extra features like 3D models, that we ignore for simplicity

The main schematic tool is called "eeschema"

- Schematic files, called "*.sch", are ASCII texts
- It relies on a library of components
  - There is a rich default library of components
  - Many components are found online, with a free license
  - You can design yours, as usual
- The output you most use is a netlist file

The PCB design tool is called "pcbnew"

- The file is called "*.kicad_pcb", and it is an ASCII text
- It relies on a library of footprints
  - There is a rich default library of footprints
  - Many footprints are found online with a free license
  - You can design yours, as usual
- The output you most use is a set of gerber files for production
  - And possibly the BOM and placement files

Tools and Common Workflow

Kicad includes two other important tools:

- "cvpcb" is used to associate footprints to components
  - This is a boring but necessary step in the design workflow
  - A well-written component already lists the few possible footprints
- "gerbview" (or "gerbv", from the geda suite) can be used to verify your output
  - Never ship a gerber without looking carefully at it

So the standard workflow looks like this:

- eeschema: you draw the schematic diagram and save the netlist
- cvpcb: it reads the netlist, you associate footprints to components
- pcbnew: it reads netlist and footprints, you route the PCB and plot gerber files
- gerbv[iew]: you check the output
- email/web: you ship the files and wait for the PCB to be delivered

Finally, mounting

- You can buy components and turn on the soldering iron
- Or, at extra cost, you can buy components, label, pack and ship for mounting
  - For this you need the BOM an position files, saved by pcbnew
- Or, at even more cost, your manufacturer does everything for you
  - You must be most precise with your BOM file

The design workflow for the lone developer

For my own projects, I prefer a different workflow

- 1- Lay out the schematics with the most important components
- 2- Draft mechanics and place core components
- 3- Add more items to the schematic diagram
- 4- Adjust mechanics if needed
- 5- Place and route the new items
- 6- Goto 3
- 7- Add anything optional that may be useful and fits mechanics
- 8- Place and route these extra things
- 9- Goto 7
- 10- Remove unused "options" before the mounting step
  - You really don't want solder alloy on your free pads

Everything must be performed with later steps in mind

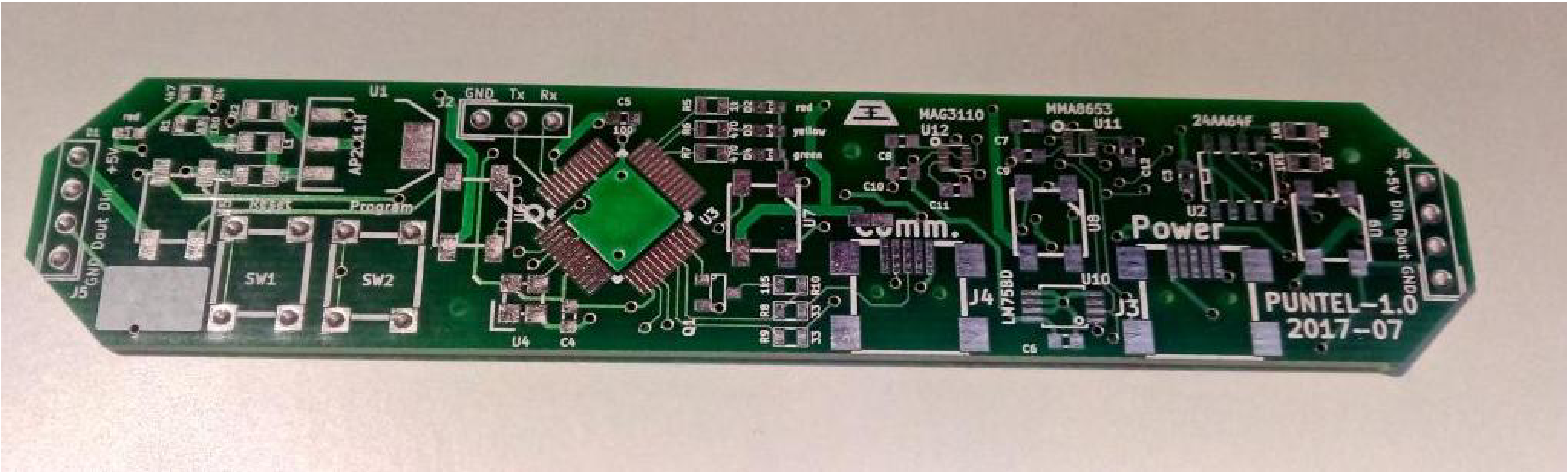
Other ideas worth considering

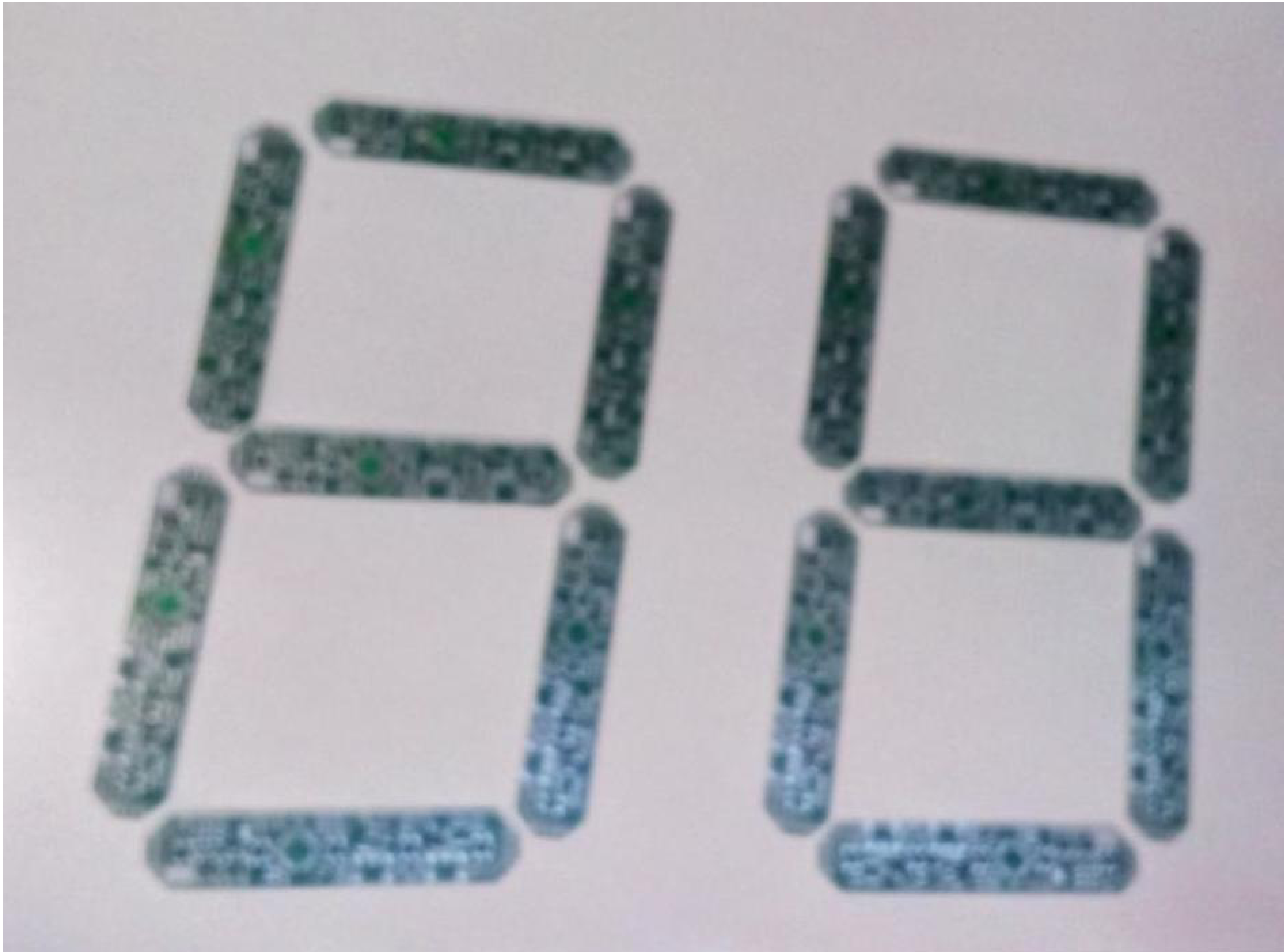
Be friendly to your software mate (who might be yourself :)

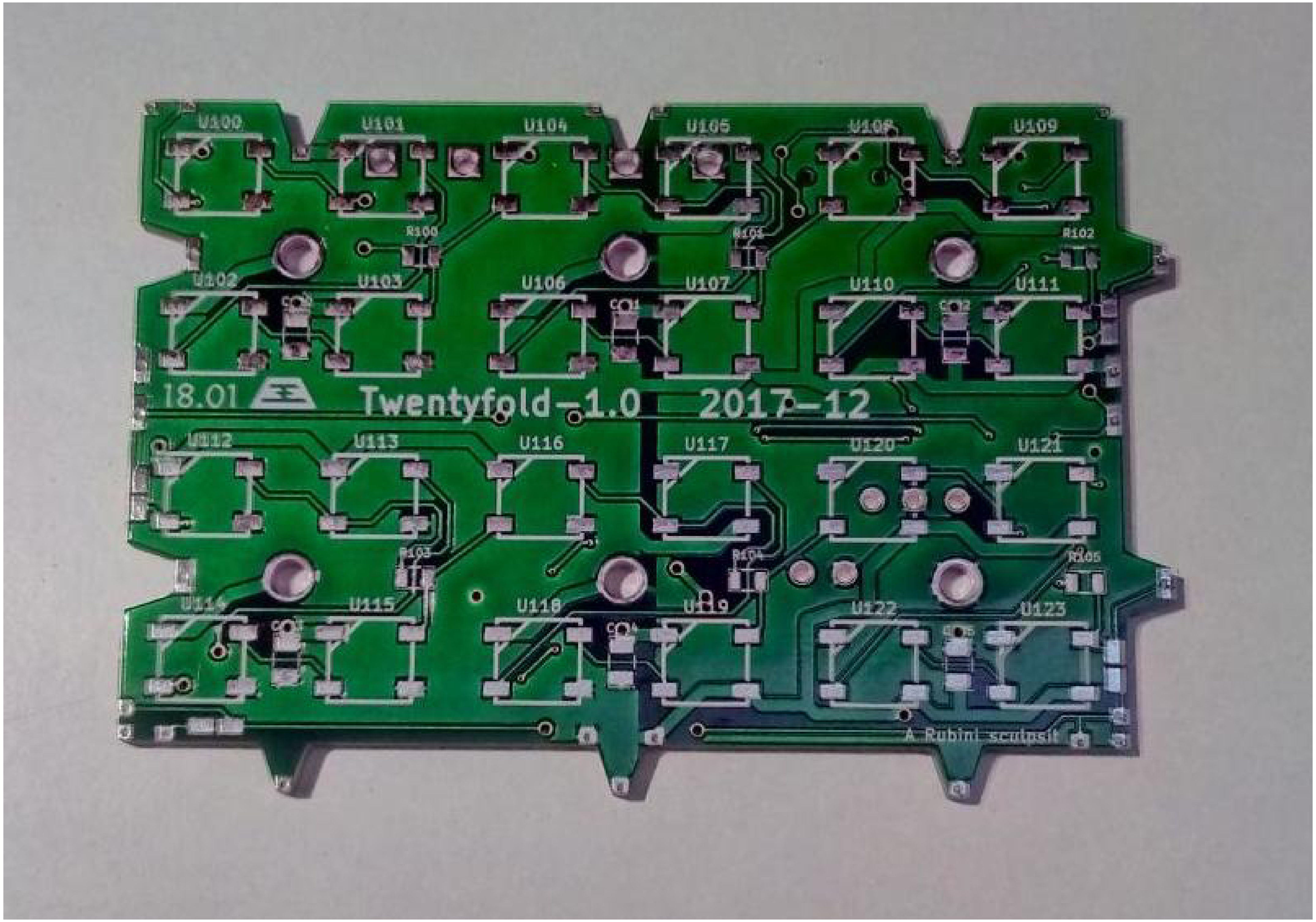
- Add I/O channels if possible
- Provide means for logging
- Add test-points whenever possible
- Remember debugging (jtag and/or something creative)
- If it fits, add new components to test
  - you might save the cost of an evaluation board later on

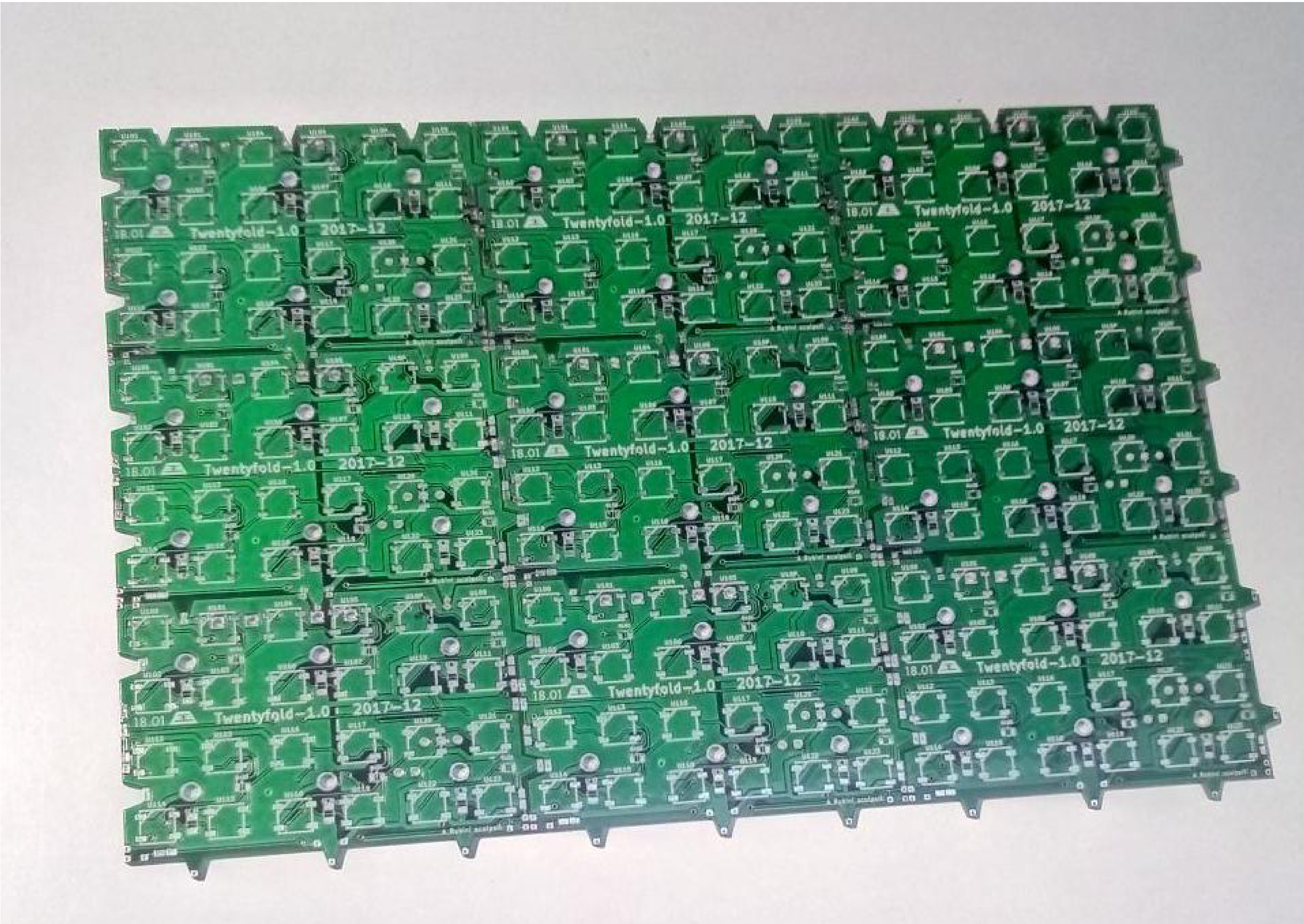
Surprise your client, by guessing their next moves

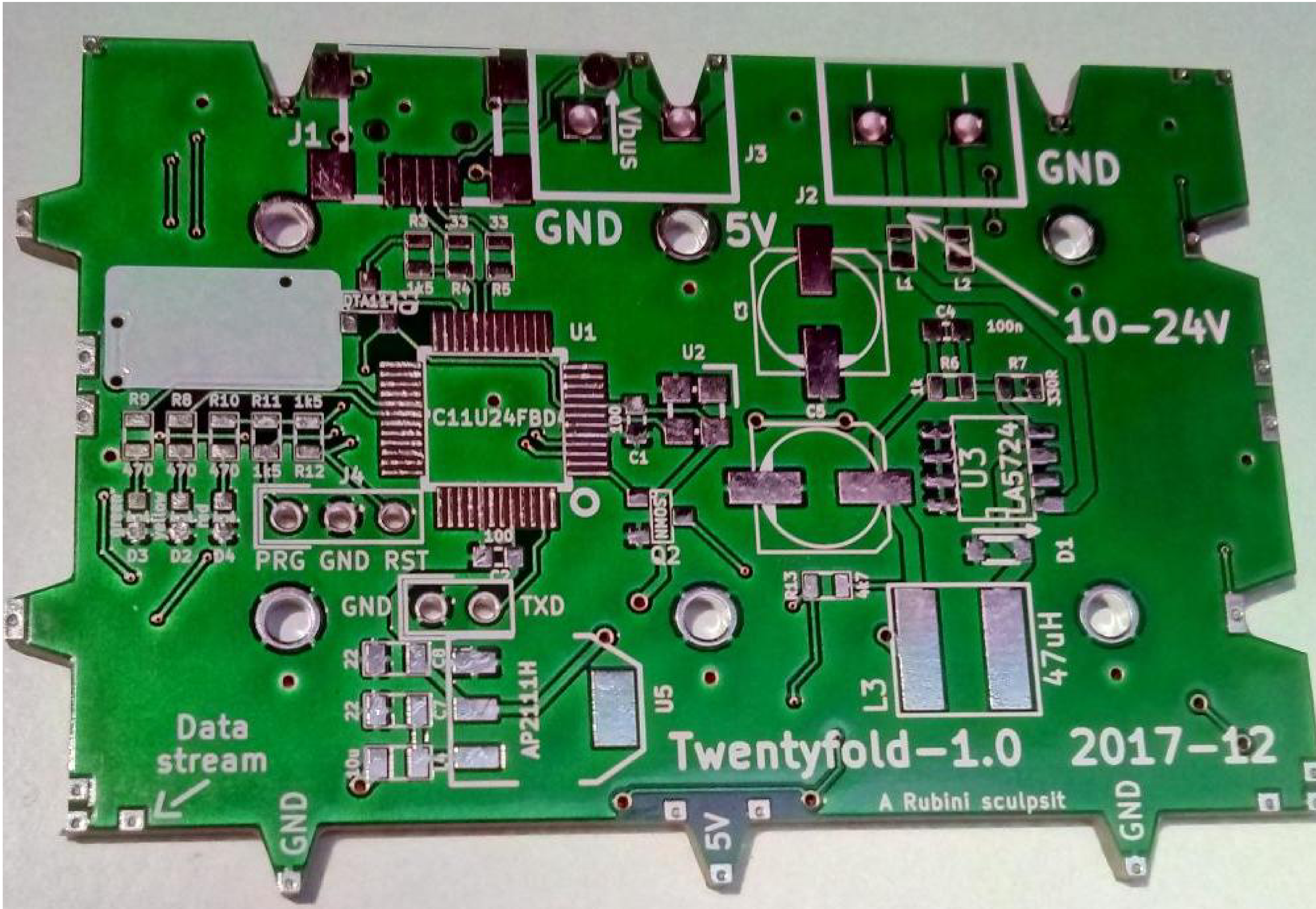
- They didn't ask for storage, but you know better
- Won't they need a thermometer sooner or later?
- Doesn't the application benefit from a light sensor?
- And why not, always offer a UEXT expansion


GND Tx Rx
J2
U1
AP2211H
Reset
Program
SW1
SW2
J5
+5V
GND Dout Din
C5
U3
U4
C4
Q1
red
yellow
green
U7
comm.
J4
MAG3110
U12
MMA8653
U11
24AA64F
U2
U8
U10
LM75BD
J3
Power
J6
+5V Din Dout GND
PUNTEL-1.0
2017-07

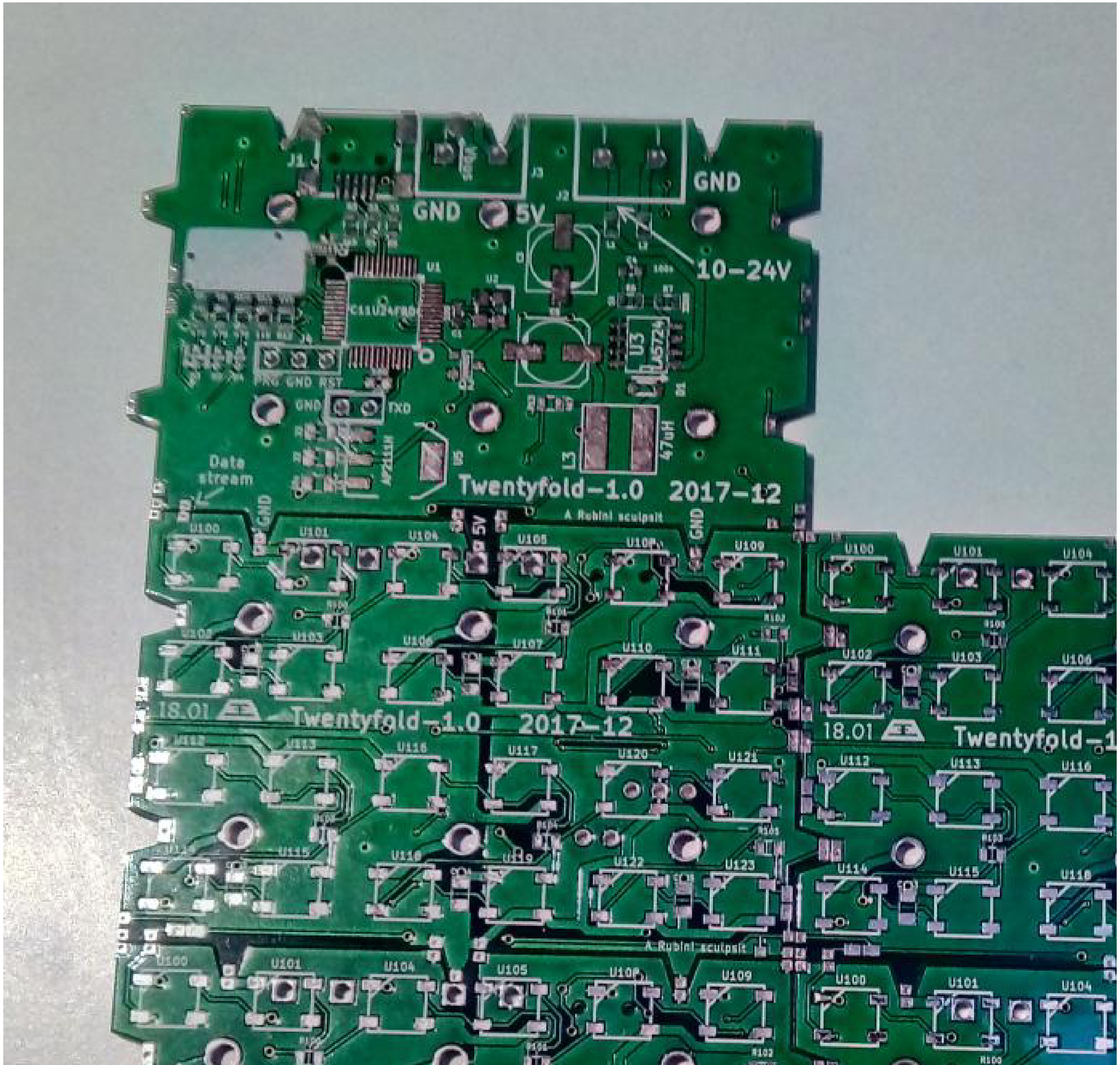


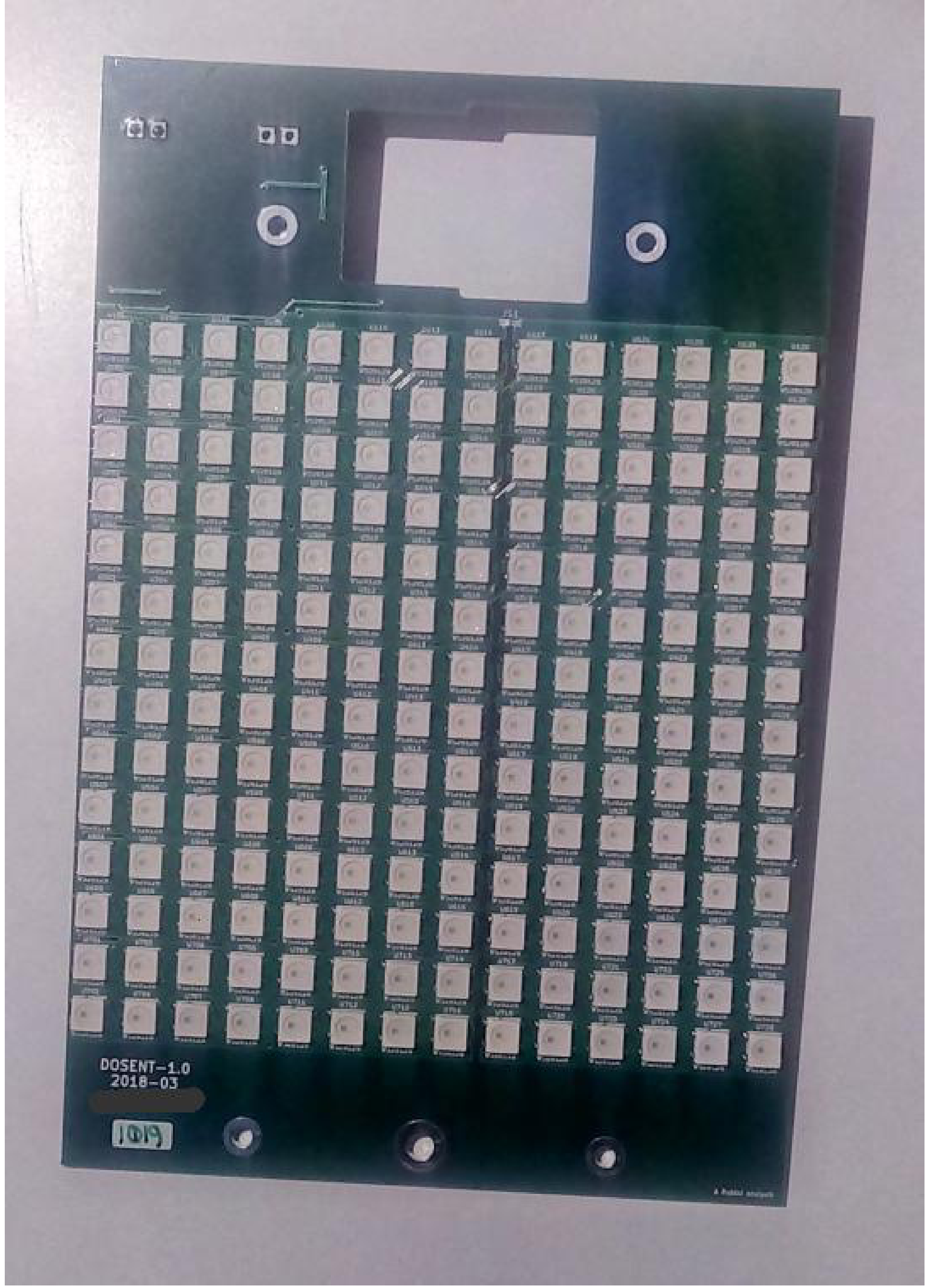

18.01
Twentyfold-1.0
2017-12
A Rubini sculpsit

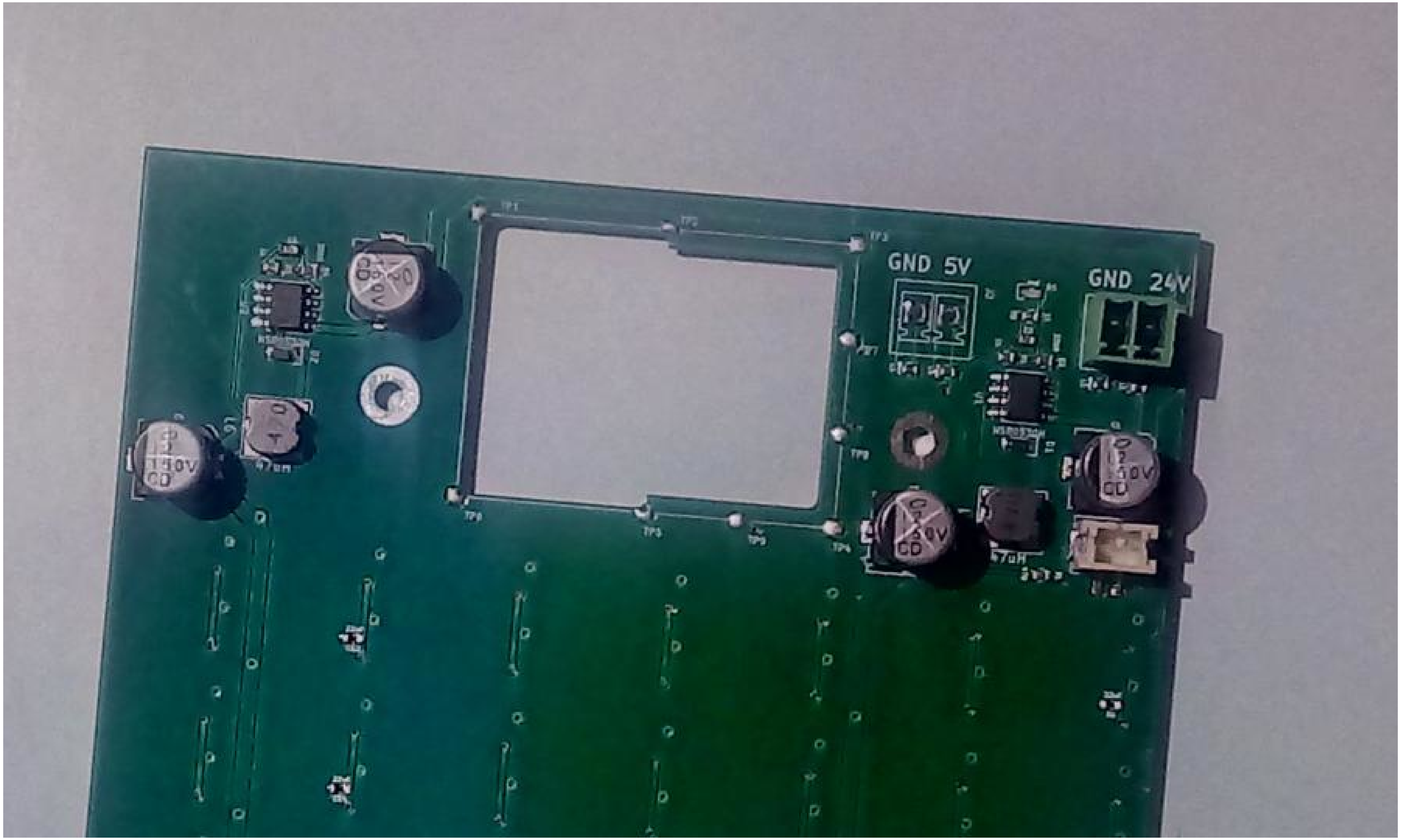

18.01
Twentyfold-1.0
2017-12

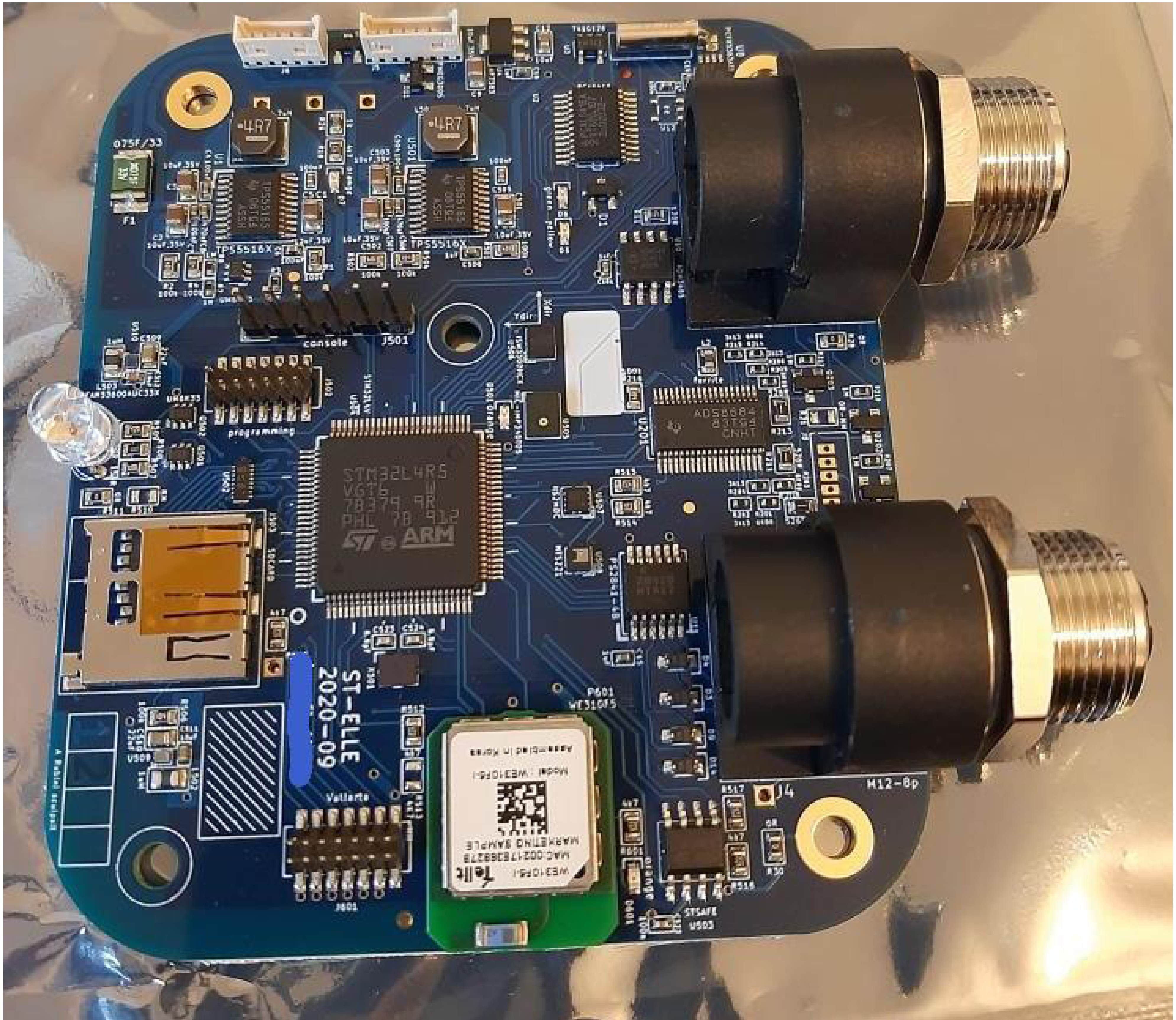


J1
J3
Vbus
GND
5V
J2
GND
10-24V
U1
U2
U3
J4
PRG GND RST
GND TXD
AP2111H
U5
L3
47uH
D1
Data
stream
Twentyfold-1.0 2017-12
A Rubini sculpsit
GND
5V
GND


GND
5V
GND
10-24V
J1
J2
J3
U1
U3
L3
47uH
PRG GND RST
GND
TXD
Data
stream
GND
5V
GND
Twentyfold-1.0
2017-12
A Rubini sculpsit
18.01
Twentyfold-1.0
2017-12
A Rubini sculpsit
18.01
Twentyfold-1


DOSENT-1.0
2018-03
[redacted]
1019


GND 5V
GND 24V
47uH



ST-ELLE
2020-09
STM32L4R5
VGT6
ARM
console
J501
programming
TPS5516X
4R7
P601
WE310F5
Assembled in Korea
Model : WE310F5-I
MARKETING SAMPLE
Telit
ADS8684
STSAFE
M12-8p
J4

But, also, there is the real world

Consider power consumption

- Your power supply must provide enough current
- Your tracks and cables must be big enough
- Are you battery-driven? Hmm... interesting....

Consider heath

- Your components get hot
- Your power supply gets hot

Consider mechanichs

- Coordinate with the mechanical engineer
- Try to ease maintainance, and component replacement

Consider noise and interference

- Switching or linear supplies?
- Beware of capacitive coupling between signals
- Beware of high impedance critical signals
- With high currents, consider the ground return path

Suggestions from a frequent user (1/2)

It's trivial stuff, but usually untold

In schematic drawing

- Make drawings small, or you'll use up too many pages (hard to manage)
- Default resistor, inductor, etc may be too big: make yours
- Use smaller-than-default text: 0.04 or 0.03 inches is best
- Use labels rather than wires, so you can change pin assignment while routing
- Add diagnostic leds for your microcontroller
- Use spare pins for board identification
- Always include a power-on led
- If it fits, add self-test circuitry
- Be generous with 0R resistors for options and tests
- Be generous with 0R1 or 1R0 resistors as amperometers
- If they fit, provide multiple alternative components

Suggestions from a frequent user (2/2)

In PCB drawing

- Place the pin1 marker outside the component
- Write component values outside the component
- Place the grid origin in a meaningful place (and mark it!)
- Always place pin-strip devices on a 2.54mm grid
- Place mounting holes at an even distance (e.g, 5mm grid)
- Add extra "mechanical" copper for all connectors
- Make pads bigger than suggested, for hand soldering
- Use square-angle routing between different layers
- Add fiducials, just in case
- Bless the board with a unique name, funny if possible
- Provide a "sticker" area for the serial number
- Proudly sign your device

My personal Suggested sizes

- References and values: 0.5mm (0.1 tick)
- Tracks: 0.2mm, 0.4mm, 0.8mm, 1.6mm
- Clearance: 0.018mm
- Grid: 1mm (or 0.5mm) for components, 0.25mm for tracks
- Vias: 0.8/0.6 or 0.5/0.3; bigger for power.