

# USB Controlled Appliances

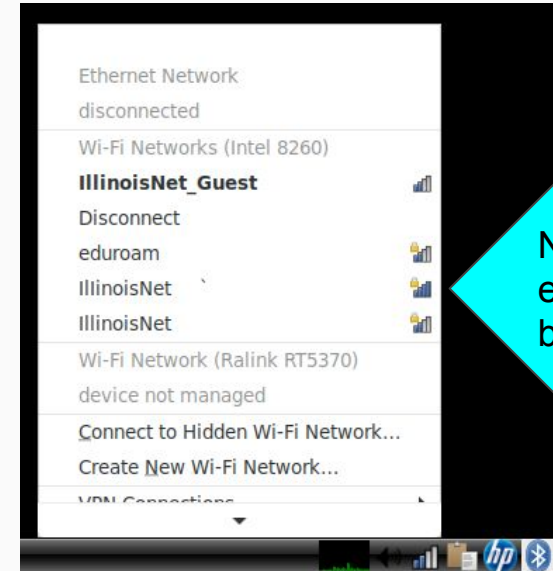
Team 15

Nagarjun Kumar, Peter Jin

Who we are

# Problem Statement

- Pretty much all IoT devices nowadays use wireless technologies.
- Attack area of home wifi networks is extremely high, to the point where it is common to segment IoT devices in a way that blocks access to more trusted devices



Note the  
extra  
backtick (`)

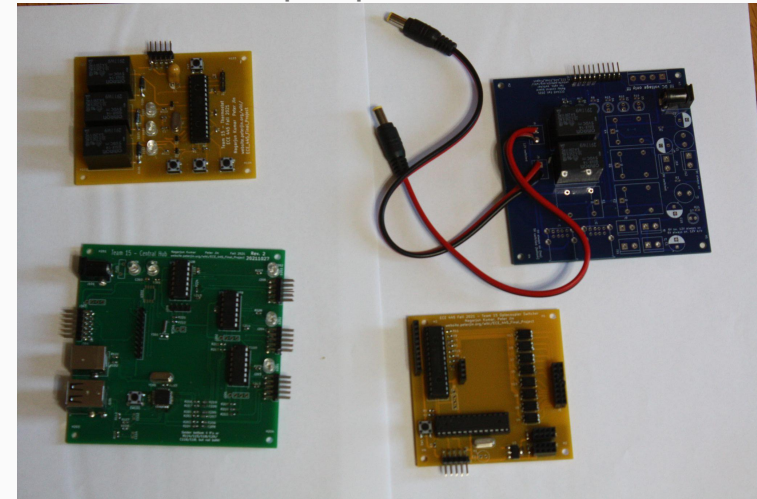
# Solution

- Make versions of existing IoT devices, but cut out the WiFi/Bluetooth/network connectivity
- Interface electrical components entirely via a computer's USB port -- quite ubiquitous nowadays!
- It's not that we don't like WiFi/Bluetooth, we're just more reserved about it



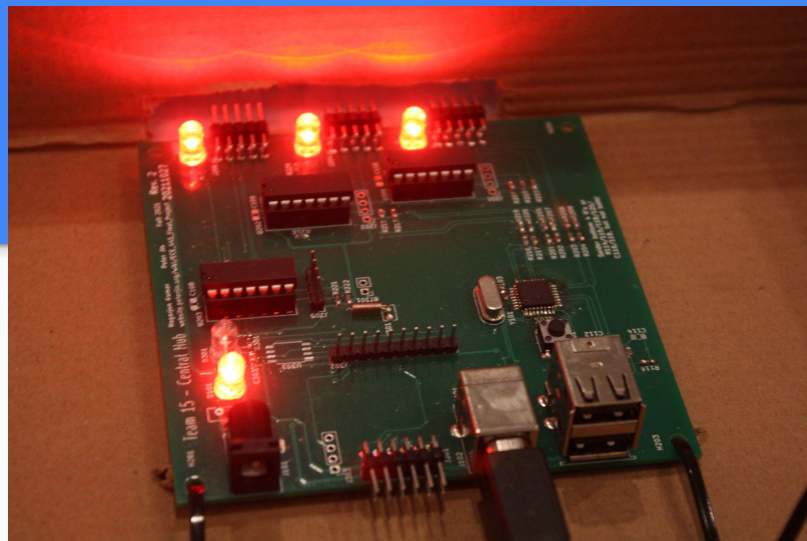
# Solution Implementation

- Many possible ways to realize this idea
  - USB interface on every IoT appliance
  - Central hub to convert one USB input to many outputs of a much simpler protocol
  - Simple protocol can be UART, I2C, or 1-Wire



# Central Hub

- One 4-port USB hub + 3x USB-to-UART adapters (MCP2221A)
- I2C bus breakout, with RTC and GPIO expander as test devices
- No microcontroller programming anywhere!



```
sebastian@tux-pc-8007:~$ sudo i2cdetect -y 9
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  26  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  57  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  6f  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

# Central Hub Requirements and Verifications

## Requirements

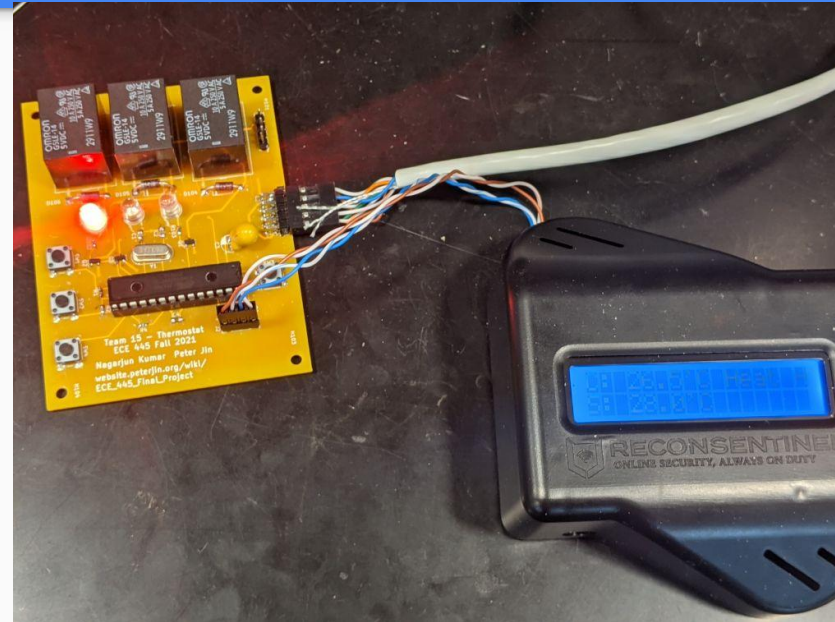
For each appliance connected to the central hub, the appliance must work regardless of which output port on the hub the appliance was plugged into.

The central hub must be able to support three devices plugged into it at the same time, and in any combination.

```
[389122.916515] usb 1-2: new full-speed USB device number 23 using xhci_hcd
[389123.080162] usb 1-2: New USB device found, idVendor=0451, idProduct=2046, bcdDevice= 1.25
[389123.080175] usb 1-2: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[389123.142269] hub 1-2:1.0: USB hub found
[389123.145148] hub 1-2:1.0: 4 ports detected
[389123.468073] usb 1-2.2: new full-speed USB device number 24 using xhci_hcd
[389123.599193] usb 1-2.2: New USB device found, idVendor=0408, idProduct=00dd, bcdDevice= 1.00
[389123.599205] usb 1-2.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[389123.599209] usb 1-2.2: Product: ECE 445 Central Hub
[389123.599213] usb 1-2.2: Manufacturer: Peter Jin, Nagarjun Kumar
[389123.746243] cdc_acm 1-2.2:1.0: ttyACM0: USB ACM device
[389123.746296] usbcore: registered new interface driver cdc_acm
[389123.746299] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
[389123.751501] hid-generic 0003:0408:0000.0003: hiddev0,hidraw2: USB HID v1.11 Device [Peter Jin, Nagarjun Kumar ECE 445 Central Hub] on usb-0000:04:00.3-2.2/input2
[389123.751570] usbcore: registered new interface driver usbbid
[389123.751574] usbbid: USB HID core driver
[389123.773378] mcp2221 0003:0408:0000.0003: hidraw2: USB HID v1.11 Device [Peter Jin, Nagarjun Kumar ECE 445 Central Hub] on usb-0000:04:00.3-2.2/input2
[389123.804020] usb 1-2.3: new full-speed USB device number 25 using xhci_hcd
[389123.935199] usb 1-2.3: New USB device found, idVendor=0408, idProduct=00dd, bcdDevice= 1.00
[389123.935212] usb 1-2.3: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[389123.935216] usb 1-2.3: Product: ECE 445 Central Hub
[389123.935220] usb 1-2.3: Manufacturer: Peter Jin, Nagarjun Kumar
[389124.041345] cdc_acm 1-2.3:1.0: ttyACM1: USB ACM device
[389124.046387] mcp2221 0003:0408:0000.0004: hidraw3: USB HID v1.11 Device [Peter Jin, Nagarjun Kumar ECE 445 Central Hub] on usb-0000:04:00.3-2.3/input2
[389140.728083] usb 1-2.4: new full-speed USB device number 26 using xhci_hcd
[389140.859146] usb 1-2.4: New USB device found, idVendor=0408, idProduct=00dd, bcdDevice= 1.00
[389140.859160] usb 1-2.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[389140.859165] usb 1-2.4: Product: ECE 445 Central Hub
[389140.859168] usb 1-2.4: Manufacturer: Peter Jin, Nagarjun Kumar
[389140.969349] cdc_acm 1-2.4:1.0: ttyACM2: USB ACM device
[389140.974672] mcp2221 0003:0408:0000.0005: hidraw4: USB HID v1.11 Device [Peter Jin, Nagarjun Kumar ECE 445 Central Hub] on usb-0000:04:00.3-2.4/input2
```

# Thermostat

- Simple thermostat with 3-wire HVAC hookup using relays
- Intended to be similar in functionality to Nest, Ecobee, etc.
- Uses a single Arduino (ATMEGA328P)
- UART interface allows button presses to be simulated and temperature to be monitored on the computer



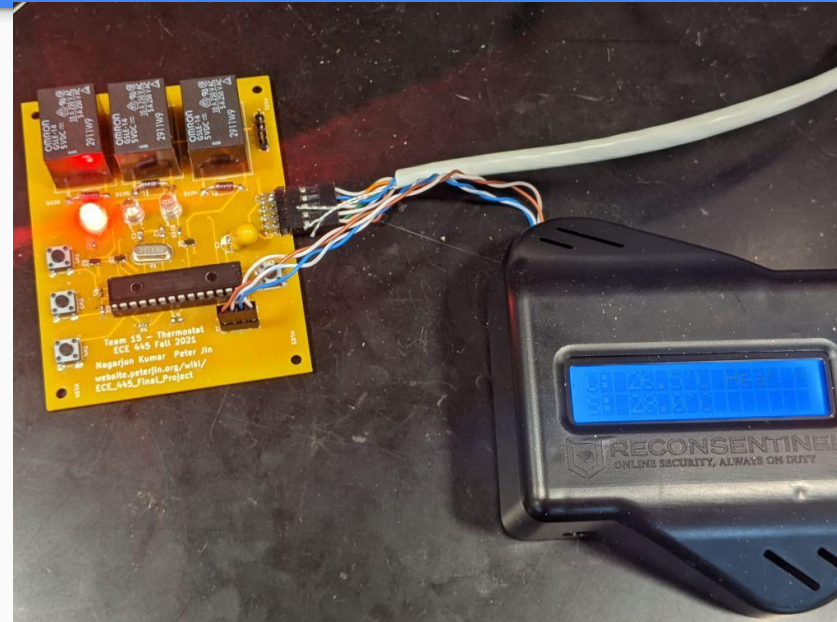


# Thermostat Requirements and Verifications

## Requirements

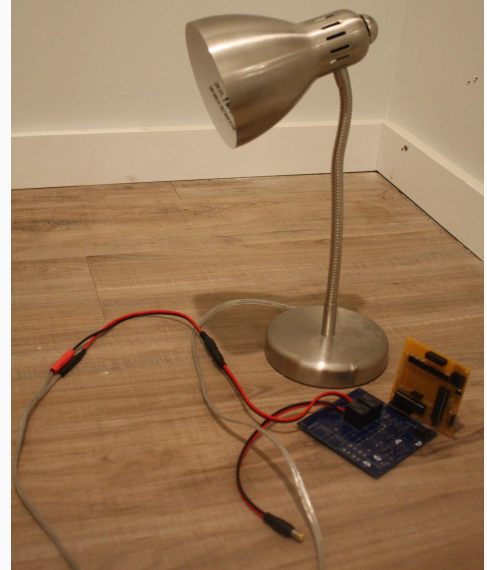
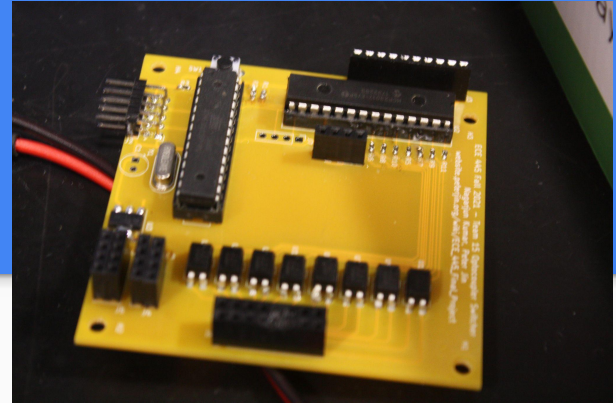
The computer is able to control the “hot”, “cold”, and “fan” outputs manually.

Buttons on the thermostat must switch between heat, cool, and fan modes, as well as increasing or decreasing the temperature.



# Optocoupler Switcher

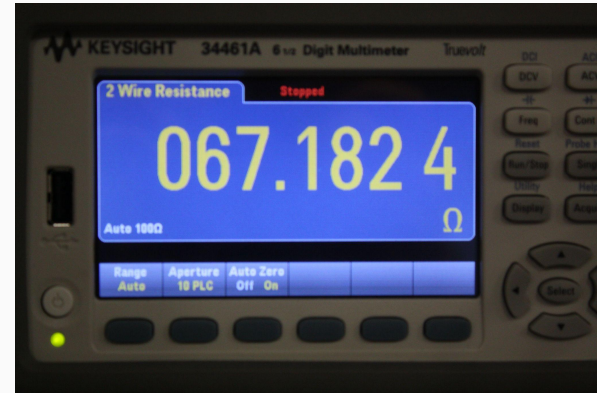
- Arduino with UART to central hub and I2C to 16-output GPIO expander
- Not very useful on its own, so we designed a relay board and connected a lamp to turn it on and off
- Also has a spot for a motion sensor/detector module



# Optocoupler Requirements and Verifications

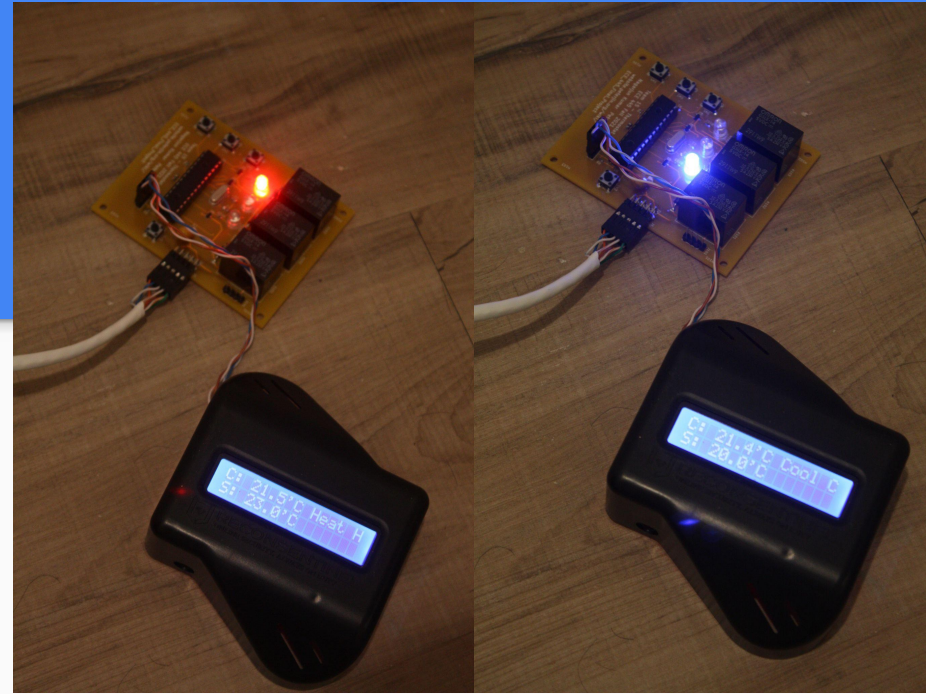
## Requirement

The optocoupler switcher appliance must work with both an active-high and active-low remote control button model.



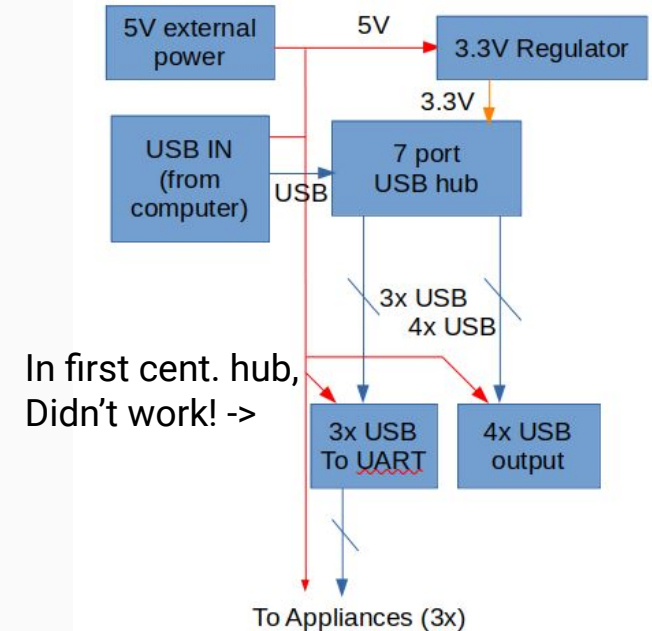
# Functional Verifications

- Central Hub
  - Fully functional.
  - Integration with three simultaneous appliances also fully functional.
- Thermostat
  - Fully functional.
- Optocoupler Switcher
  - Appears to be fully functional.
  - GPIO outputs work when connected to the relay board
  - Optocoupler outputs only tested with resistance measurement between C and E
  - Motion detector was unfortunately erratic, but the GPIO state of the motion detector module was still read correctly by the ATMEGA328P



# Failures

- Initial version of central hub had a 7-port USB hub
  - Rationale: Allow more instances of the central hub to be connected together without having to daisy chain
  - Unfortunately, the 7 port USB hub only detected the first 4 ports; the USB-to-UARTs are on ports 5, 6, and 7
- Optocoupler switcher was totally not working in initial iteration
  - Incorrect wiring of ATMEGA328P reset button, and microcontroller wouldn't start or accept code
- In the end, everything still worked!



# Future Work

- Creating more types of appliances
  - Washers, dryers, refrigerators, security systems, door locks
  - Ethernet switch VLAN management interface controller
- Generic integration with existing IoT devices
  - Concept already created: IoTether - <https://website.peterjin.org/wiki/IoTether>
  - 4 port USB 2.0 high-speed hub with Ethernet controller
  - Allows us to have fine-grained control of the network connectivity of an arbitrary IoT device, by plugging in a switch and/or wireless access point into the Ethernet controller
  - MCP2221A attached to the high-speed hub would control power or reset buttons using the mechanism from the optocoupler switcher or allow access to debug UARTs on the embedded systems
  - Incorporating HVAC system to thermostat to be able to control temperature, AC, heat, and fan.

# Ethics

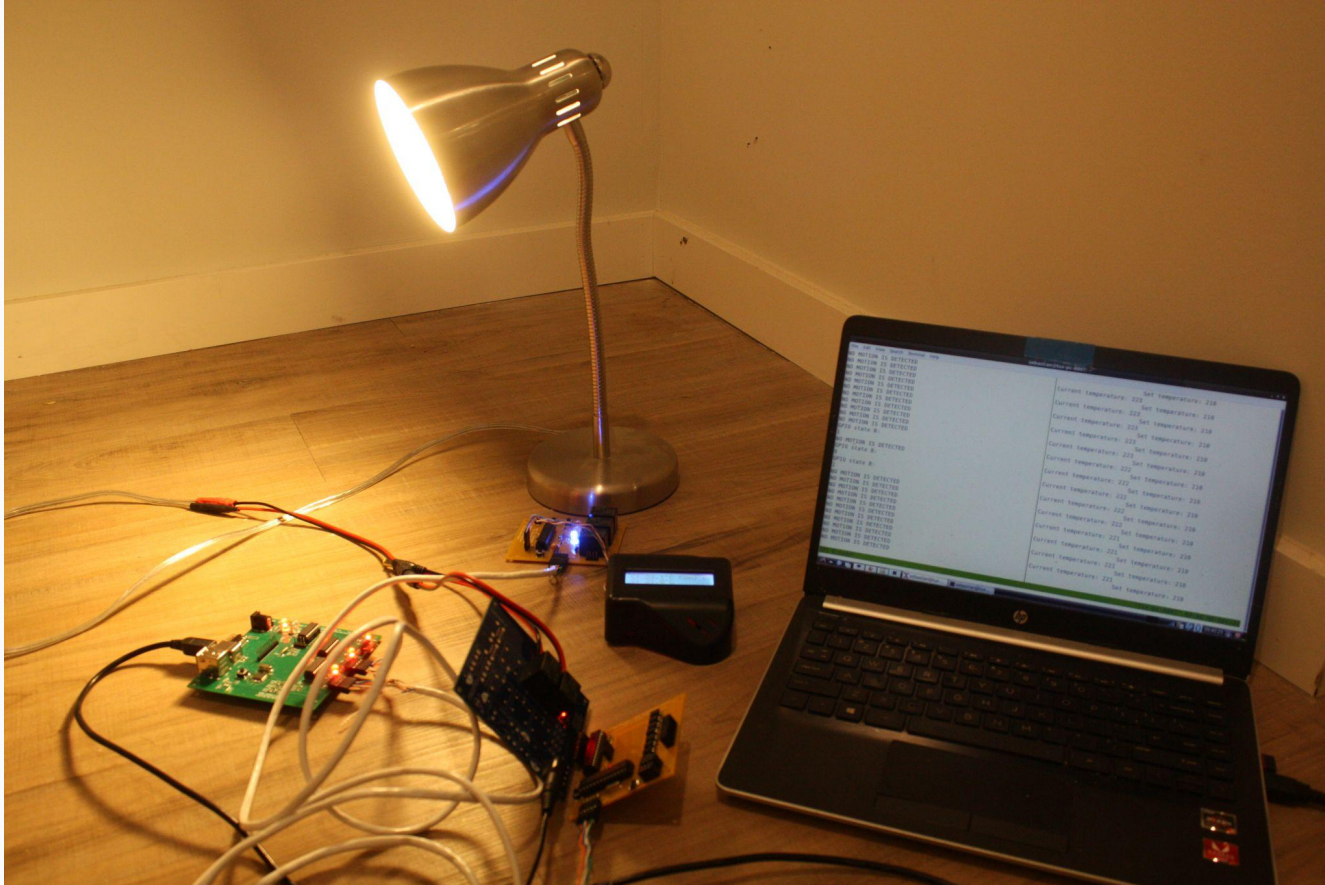
- Ultimately intended to be useful for privacy-minded people, not for the average consumer
  - Important to be open and transparent, and use open-source software wherever possible
- We decided on the MCP2221A mostly because of its good Linux support for the I2C and UART
  - Driver for the MCP2221A (“hid\_mcp2221”) already present in mainline Linux kernel
  - No closed-source binary blobs to run on the computer
- PCB design released as open source / open hardware
  - <https://git2.peterjin.org/usb-controlled-appliances>

# Conclusion

- Learned about interaction between code and hardware.
- Benefits of different types of standard protocols (UART, I2C, USB).



Thank you!



# Further Reading

- [https://website.peterjin.org/wiki/USB\\_Controlled\\_Appliances](https://website.peterjin.org/wiki/USB_Controlled_Appliances)