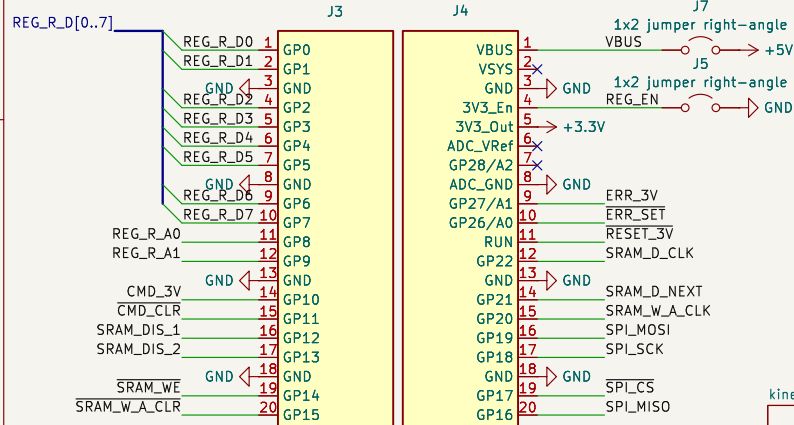


The 3.3V regulator built into the microcontroller is documented as only supplying a max of 300mA to external circuitry.

Rather than use that, we use our own regulator in the cartridge board. It can power the microcontroller and all 3V logic chips. Then we power the microcontroller through the 3V3\_OUT and GND pins.

One jumper connects 3V3\_EN connects to GND to disable the board's regulator. Another jumper connects VBUS to the +5V supply to power the 5V chips.

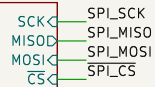
While programming the microcontroller in-place via USB, the jumper should be on the VBUS pins. While operating in the Sega, the jumper should be on the 3V3\_EN pins.



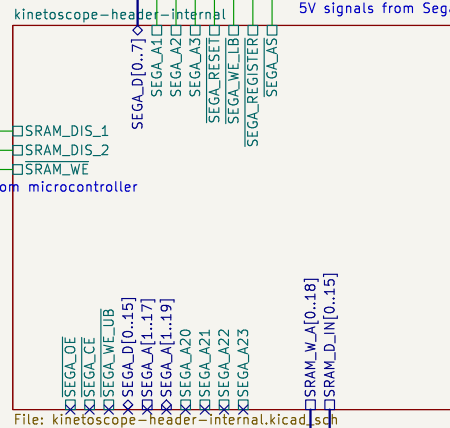
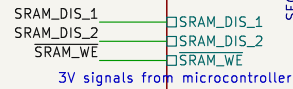
X2

Template for RPiPicoW board header alignment

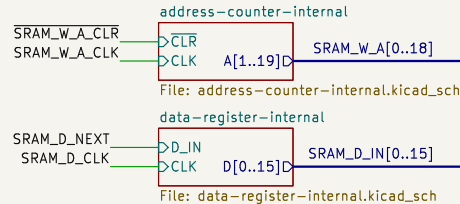
ethernet-internal



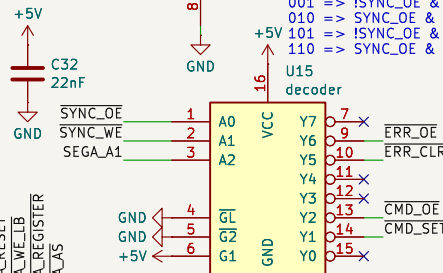
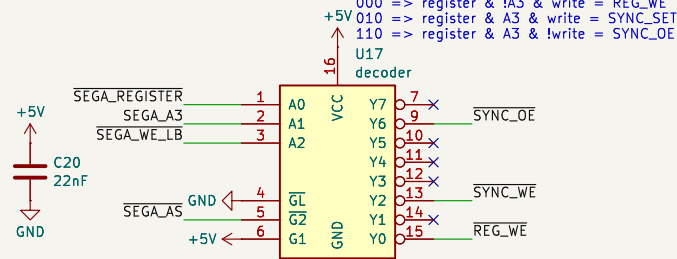
File: ethernet-internal.kicad\_sch



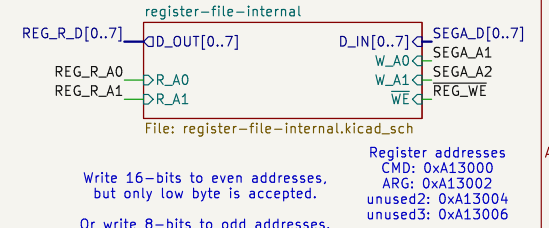
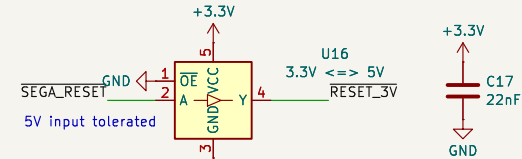
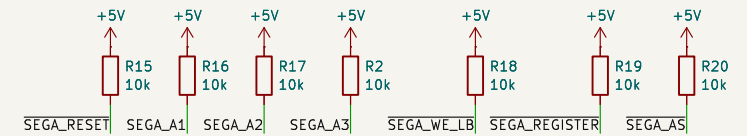
File: kinetoscope-header-internal.kicad\_sch



File: data-register-internal.kicad\_sch



5V signals from Sega



Register addresses  
CMD: 0xA13000  
ARG: 0xA13002  
unused2: 0xA13004  
unused3: 0xA13006

Command token address  
0xA13008

Error token address  
0xA1300A

Write 16-bits to even addresses,  
but only low byte is accepted.  
Or write 8-bits to odd addresses.

000 => register & !A3 & write = REG\_WE  
010 => register & A3 & write = SYNC\_SET  
110 => register & A3 & !write = SYNC\_OE

001 => !SYNC\_OE & SYNC\_WE & !A1 = CMD\_SET  
010 => SYNC\_OE & !SYNC\_WE & !A1 = CMD\_OE  
101 => !SYNC\_OE & SYNC\_WE & A1 = ERR\_CLR  
110 => SYNC\_OE & !SYNC\_WE & A1 = ERR\_OE

5V signal CMD\_SET

3V signal CMD\_CLR

5V signal CMD\_OE

3V signal ERR\_SET

5V signal ERR\_CLR

5V signal ERR\_OE

While powering the microcontroller via USB without the Sega (during programming and testing), pull-up resistors on these lines keep inputs from floating. They can be set low for testing with jumper wires in the cartridge pin header.

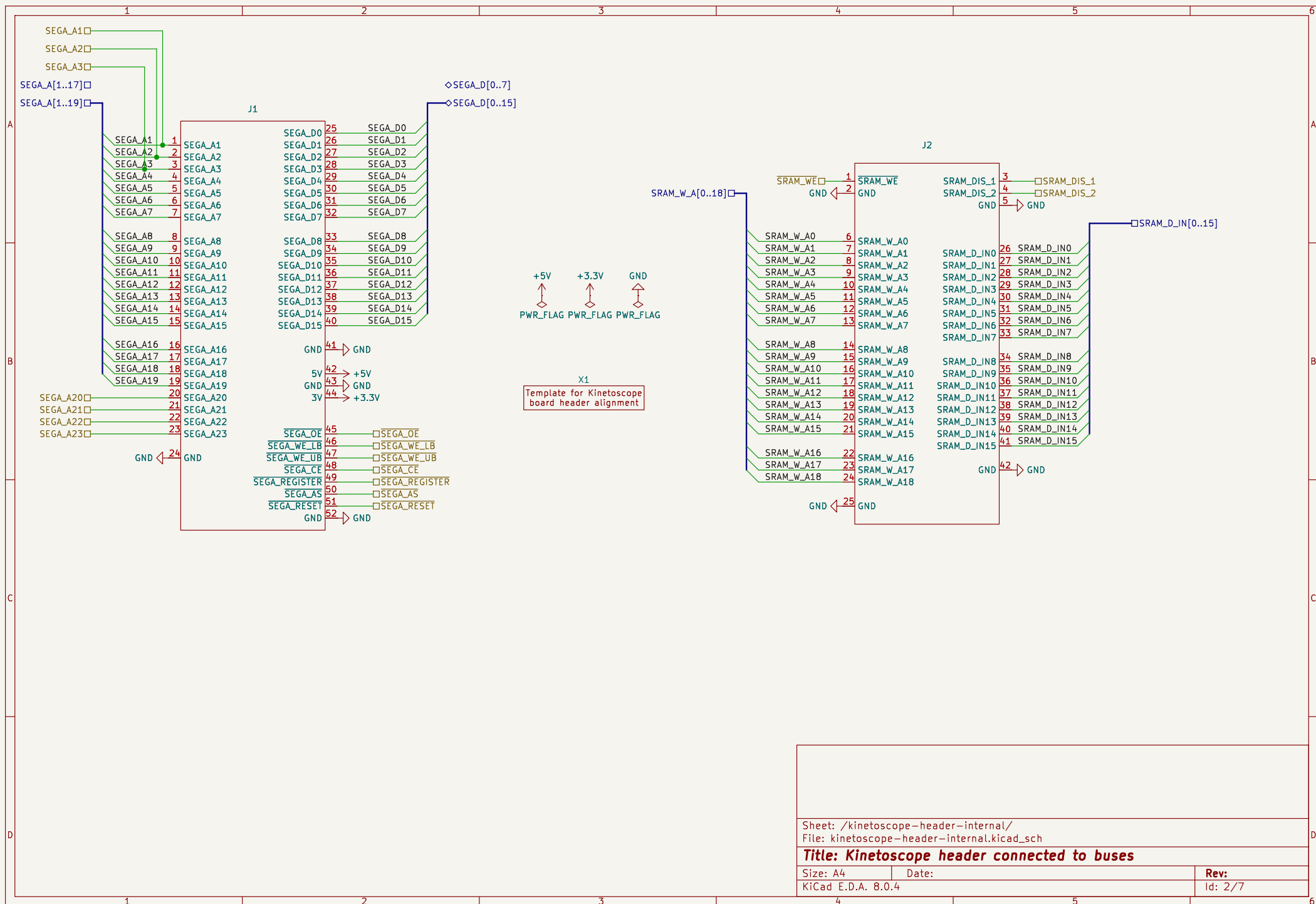
Sheet: /  
File: microcontroller.kicad\_sch

**Title: Kinetoscope Microcontroller and Register Board**

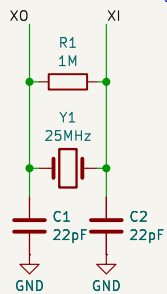
Size: A4  
KiCad E.D.A. 8.0.4

Date:

Rev:  
Id: 1/7



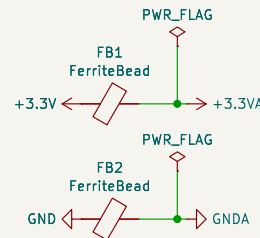
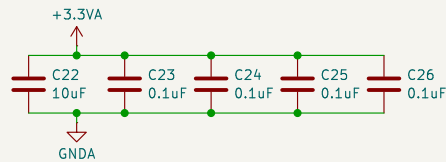
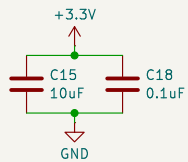
The clock circuit, the filter caps, and everything related to the ethernet jack are copied from Adafruit's Ethernet Featherwing design.



IRQ and reset signals are unused.

Pulling all three PMODE lines high enables ethernet autonegotiation.

Datasheet states that pin 23 (RSVD) must be tied to ground.

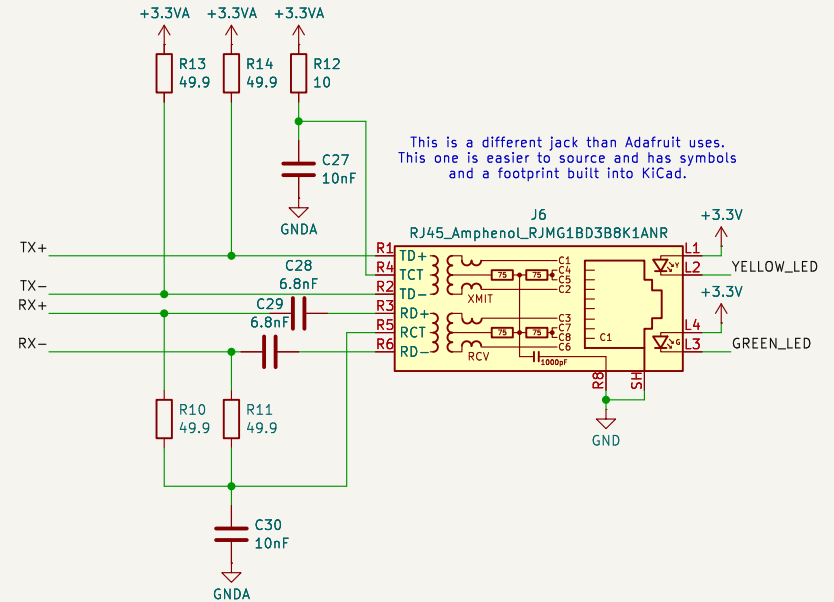
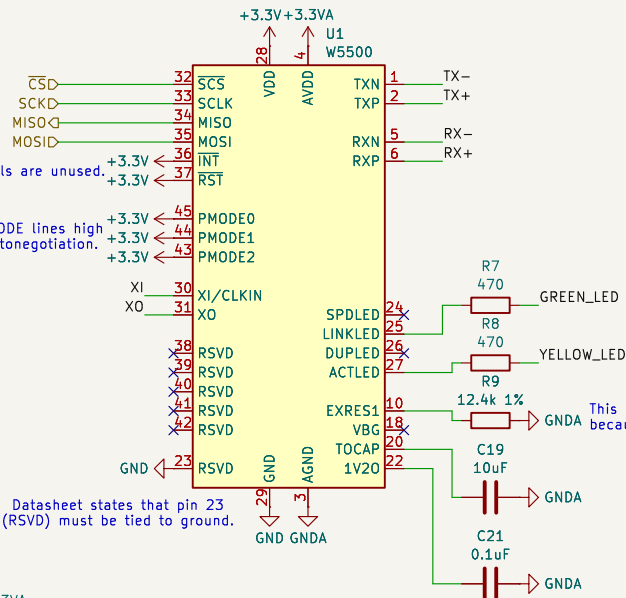


Sheet: /ethernet-internal/  
File: ethernet-internal.kicad\_sch

# **Title: Kinetoscope Ethernet Module**

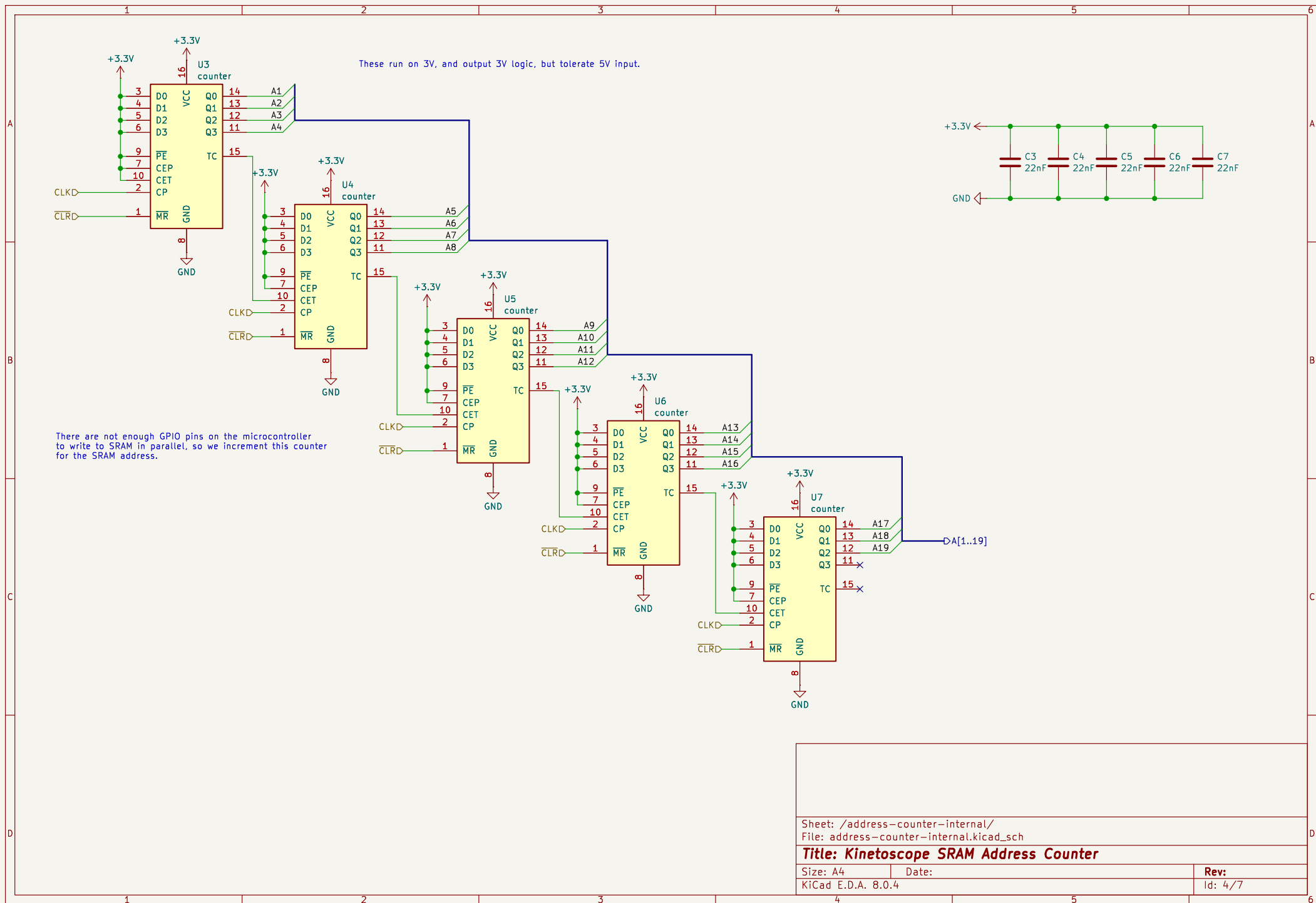
Size: A4  
KiCad E.D.A. 8.0.4

Date:  
Rev:  
Id: 3/7

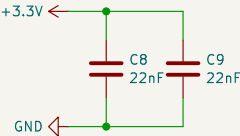
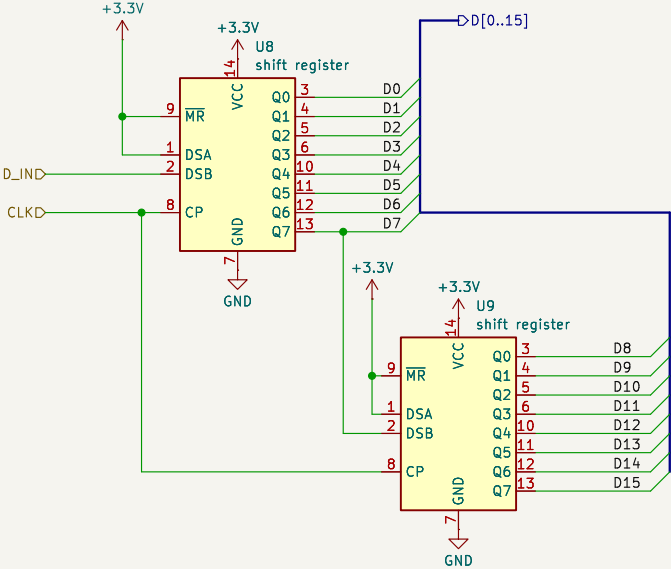


This is a different jack than Adafruit uses. This one is easier to source and has symbols and a footprint built into KiCad.

This resistor needs to be this value and precise to 1%, because W5500 uses it to bias its internal analog circuits.



There are not enough GPIO pins on the microcontroller  
to write to SRAM in parallel, so we fill this register serially.



Sheet: /data-register-internal/  
File: data-register-internal.kicad\_sch

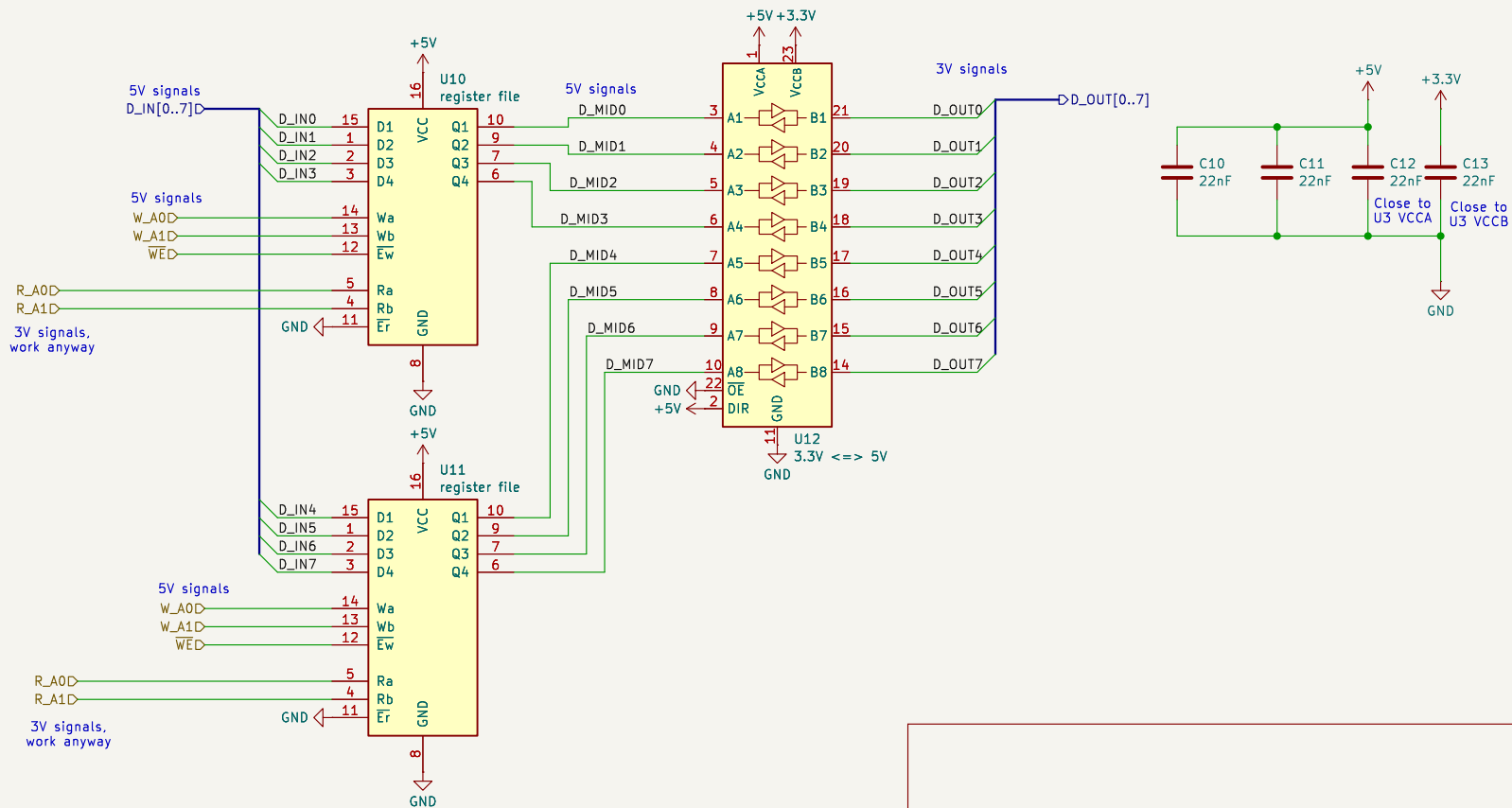
**Title: Kinetoscope SRAM Data Register**

Size: A4	Date:	Rev:
KiCad E.D.A. 8.0.4		Id: 5/7

This is a set of four 8-bit registers that can be independently read and written by the two controllers (one reads, one writes).

The registers are assumed to be written by a device with a shared data bus (Sega Genesis) and read by a device with dedicated pins (microcontroller). The read output is always enabled.

Everything runs on +5V, but must be compatible with 3.3V inputs. If using a slightly different part number, please check input voltages in the data sheet. The data outputs are 3.3V.



Sheet: /register-file-internal/  
File: register-file-internal.kicad\_sch

**Title: Kinetoscope Register File connected to buses**

Size: A4

Date:

Rev:

KiCad E.D.A. 8.0.4

Id: 6/7

This is a synchronization token between the M68k and the microcontroller.

The token will be cleared on boot/reset.

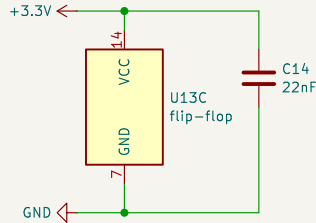
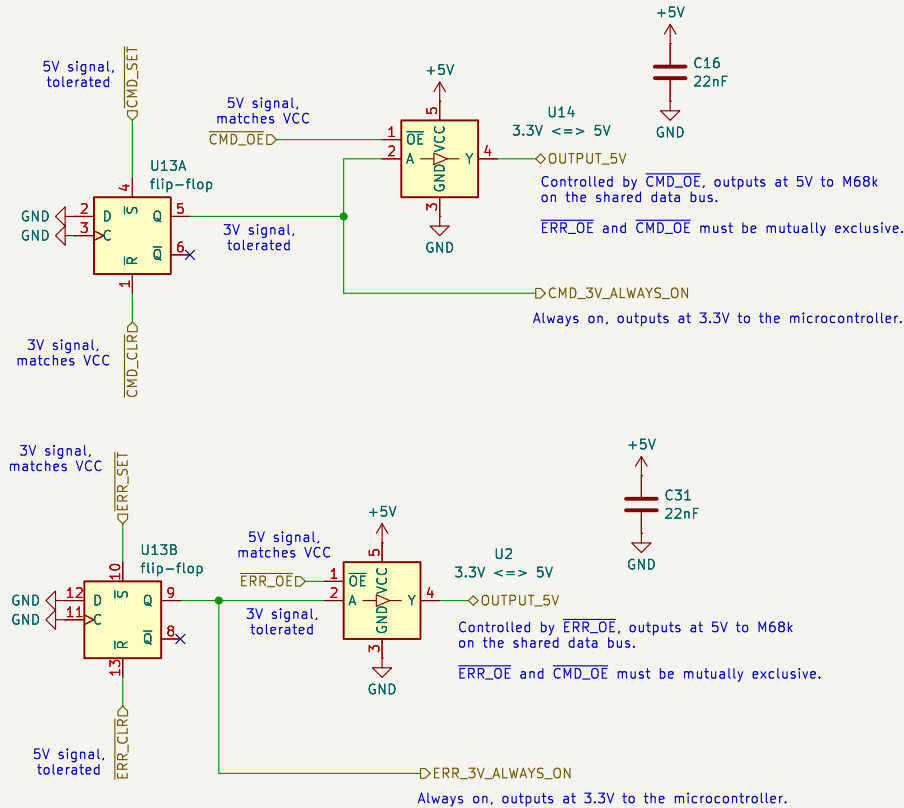
The M68k will be able to set the token ( $\overline{\text{CMD\_SET}}$ ), indicating to the microcontroller that a command is ready to be executed.

The microcontroller will be able to clear the token ( $\overline{\text{CMD\_CLR}}$ ), indicating to the M68k that the command has been executed.

The M68k and microcontroller both will be able to read the current state of the token. The M68k reads a 5V version of the signal, and the microcontroller reads a 3.3V version of the signal.

Command codes and arguments are passed through a separate register file.

The microcontroller can also flag an error  $\overline{\text{ERR\_SET}}$  to the M68k, who can clear it with  $\overline{\text{ERR\_CLR}}$  after noticing.



Sheet: /sync-token-internal/  
File: sync-token-internal.kicad\_sch

**Title: Kinetoscope Sync Token**

Size: A4  
KiCad E.D.A. 8.0.4

Date:

Rev:  
Id: 7/7