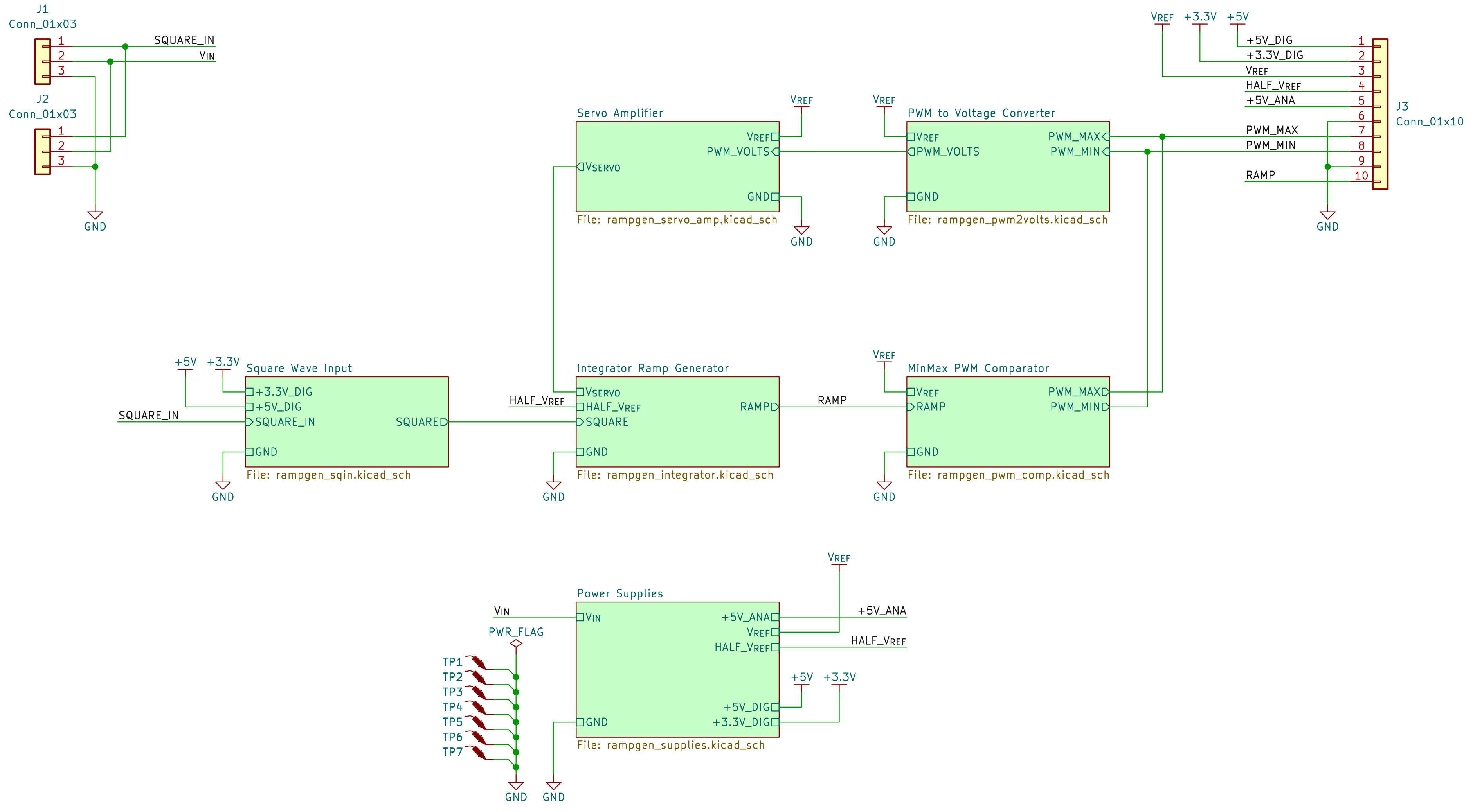


Sheet: /
File: PWM Ramp Gen Prototype.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 7.0.5		Id: 1/7

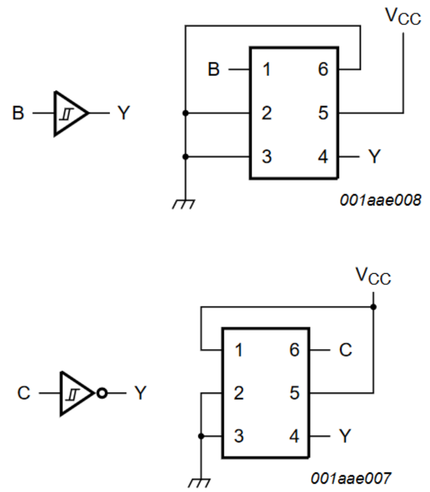
Part numbers for these fiber optic components (and the differences between them) are a little bit tricky to parse. It looks like there are a couple different generations of parts, different temperature ranges, and parts with different features (RSSI, TTL outputs, etc.)

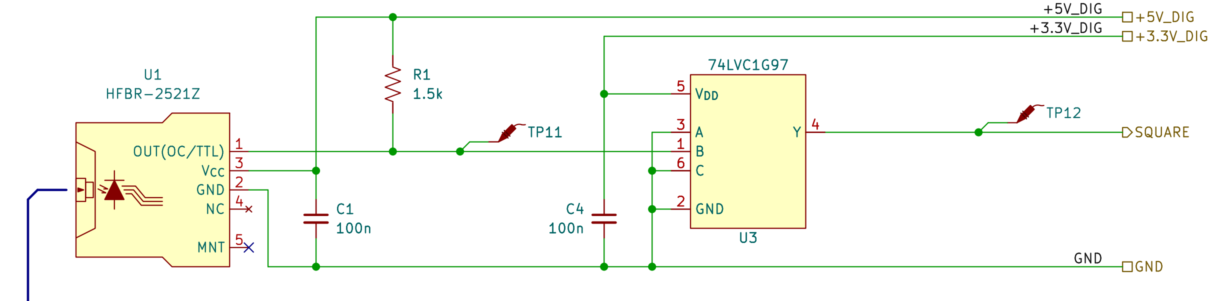
For this application, the following part no's from the HFBR-x5xxZ line should be acceptable
- HFBR-x521Z
- HFBR-x526Z?
- HFBR-x528Z

From the HFBR-x5xxETZ (extended temperature range) series:
- HFBR-x521ETZ

From the AFBRx5/6xx_Z (TTL output, RSSI) series:
- AFBR-x5/62xZ
- AFBR-x5/62xIZ

There's a couple different part numbers that may be relevant (e.g. AFBR-15x8) but it's so difficult to parse their differences so idk.
I've designed this interface circuit in an attempt to be as generic and part-agnostic as possible, but make sure to verify circuit functionality





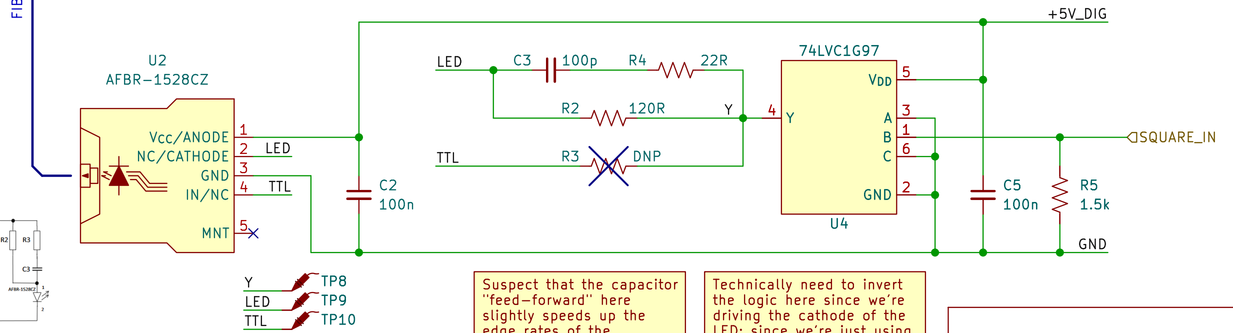
Parsed xFBR Datasheets, and I think this should be a universal pinout for these devices

Should give us flexibility in choosing which part to use exactly

Additionally, the different datasheets recommend doing different things from the N.C. pins--some recommend grounding, others recommend leaving them open.

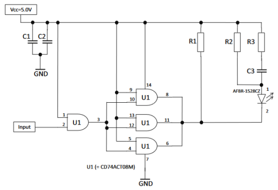
I'm going to leave them open in order to minimize any loops formed by these conductors; may have a *slight* cost of noise but flagging that for the future.

Pulling the fiber optic transceiver open collector output to 3.3V instead of 5V--respects the V_{IL} and V_{IH} of the following logic level translator slightly better than a 5V supply would (especially at lower switching frequencies)



Suspect that the capacitor "feed-forward" here slightly speeds up the edge rates of the transmitter and compensates a bit for the LED capacitance

Technically need to invert the logic here since we're driving the cathode of the LED; since we're just using this as a square wave input, this doesn't matter too much



NOTE: $I_{F,on}$ = 30 mA nominal at T_A = 25°C; V_{CC} = 5.0V.

	Value	Tolerance
R1	2 kΩ	5%
R2	100Ω	1%
R3	15Ω	1%
C1	0.1 µF	20%
C2	10 µF	20%
C3	120 pF	5%

Sheet: /Square Wave Input/
File: rampgen_sqin.kicad_sch

Title:

Size: A4 | Date: | **Rev:**

KiCad E.D.A. kicad 7.0.5 | Id: 2/7

This capacitance must be carefully managed—VSERVO is generated by an op-amp, and this capacitance forms a pole with the op-amp's output resistance.

This pole should be a lot faster than the unity gain frequency of the feedback loop (aiming for ~50Hz), but make sure to be careful here.

This inverter is useful because it allows for up/down logic level translation.

This is useful since the input square wave signal will be a fixed 3.3V logic level signal, and the output square wave will have a voltage set by VSERVO.

A feedback loop tweaks VSERVO such that a triangle wave with the appropriate amplitude is created regardless of frequency.

RC high-pass does some weird things to the transfer function, so don't make this too slow

Aiming to be equal or faster than the RC filter in the PWM to Voltage converter stage

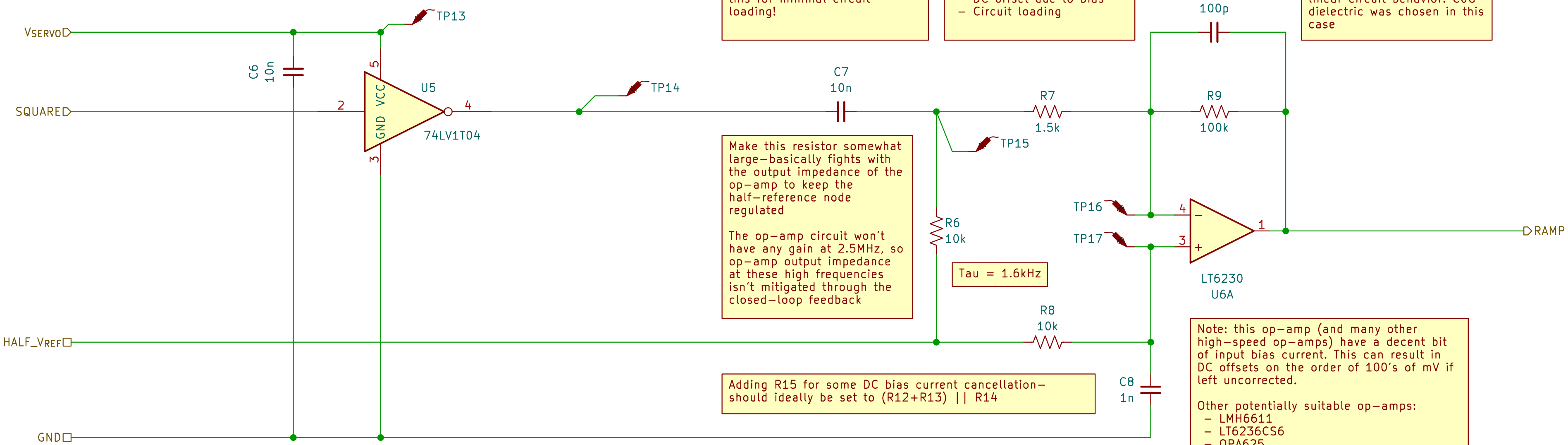
Integrator sized to produce a 0.3 to 3V triangle waveform, when a 4.25V 2.5MHz square wave is driven into the circuit input.
In the positive half of the square wave, 4.25/2V is present across the integration resistor (R13). This equates to a current flow of 1.4mA.

In the positive half of the square wave, 1.4mA * 200ns of charge (280pC) are transferred. For the output to slew 2.7V a capacitor sized approximately 100pF should be used (dQ = C * dV)

At 1MHz, capacitor has 16 ohms of impedance—make sure integration resistor has impedance much higher than this for minimal circuit loading!

Integration resistor sizing is a trade-off between
- Sensitivity to parasitics
- Thermal noise
- DC offset due to bias
- Circuit loading

Integration capacitor must be a good dielectric with reasonable precision in order to ensure predictable, linear circuit behavior. C0G dielectric was chosen in this case



Make this resistor somewhat large—basically fights with the output impedance of the op-amp to keep the half-reference node regulated

The op-amp circuit won't have any gain at 2.5MHz, so op-amp output impedance at these high frequencies isn't mitigated through the closed-loop feedback

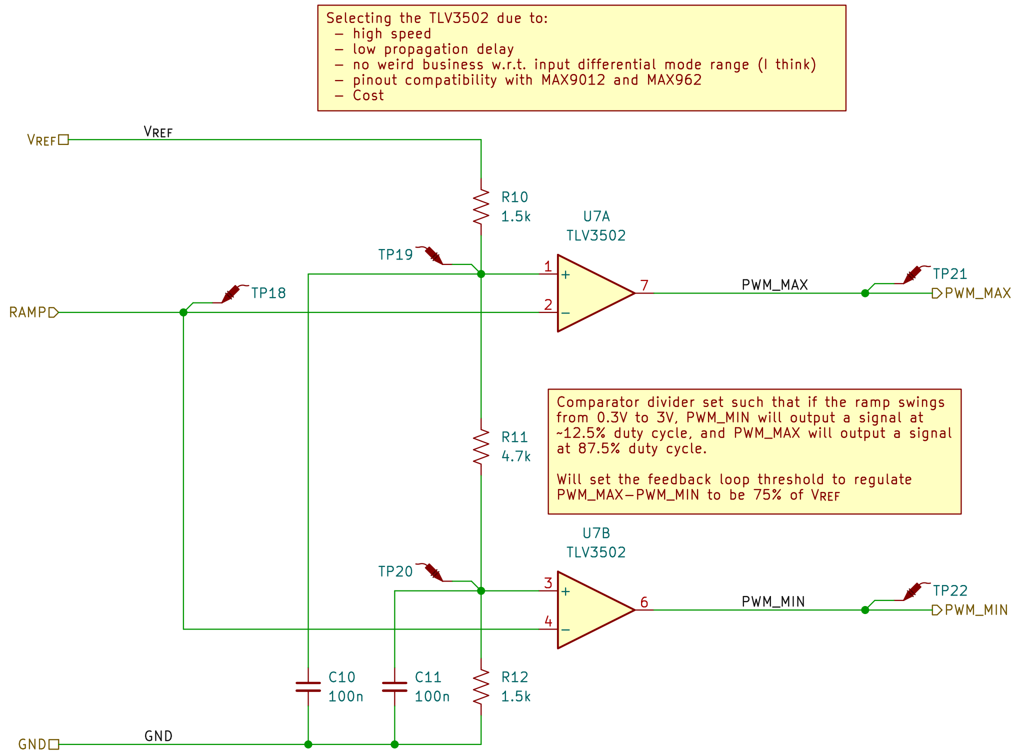
Tau = 1.6kHz

Note: this op-amp (and many other high-speed op-amps) have a decent bit of input bias current. This can result in DC offsets on the order of 100's of mV if left uncorrected.

Other potentially suitable op-amps:
- LMH6611
- LT6236CS6
- OPA625

Adding R15 for some DC bias current cancellation—should ideally be set to (R12+R13) || R14

C20 aims to filter some of the ramp frequency at the DC voltage reference node. Not adding too much capacitance in order to maintain stability of the Half-VREF op-amp



Sheet: /MinMax PWM Comparator/
File: rampgen_pwm_comp.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 7.0.5		Id: 4/7

Depending on the peak–peak ramp voltage coming out of the ramp generator, PWM_MIN and PWM_MAX will be set to certain duty cycles.

We want to regulate the difference between the MIN and the MAX duty cycles ––> this corresponds to a particular voltage magnitude of the ramp function.
NOTE: ramp voltage –> PWM delta is a monotonic function, i.e. every ramp voltage value corresponds to a unique PWM delta.

In order to implement this analog regulator, we'll first create a waveform representing the difference between the PWM waveforms using a multi–function logic gate. We'll then take the average value of this synthesized PWM waveform using an RC lowpass filter. This will yield us a voltage representing the difference bewteen the high and low PWM values

IMPORTANT NOTE: Instead of converting the "valid" PWM region to an analog voltage, I'm converting the "INVALID" region to an analog voltage.

Both of these transfer functions are nonlinear (have V_{servo} in the denominator). However, the PWM_INVALID signal has a small–signal gain that's strictly positive. This is preferable because implementing the controller in the analog domain is significantly easier for forward path transfer functions with positive gains.

In order to generate the required PWM_INVALID waveform, we'll use a multi–function gate. The output waveform is HIGH iff:
– PWM MAX is LOW
–– or ––
– PWM MIN is HIGH
As such, an OR gate with a single inverting and non inverting input can accomplish this task

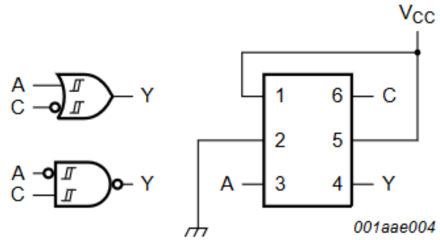
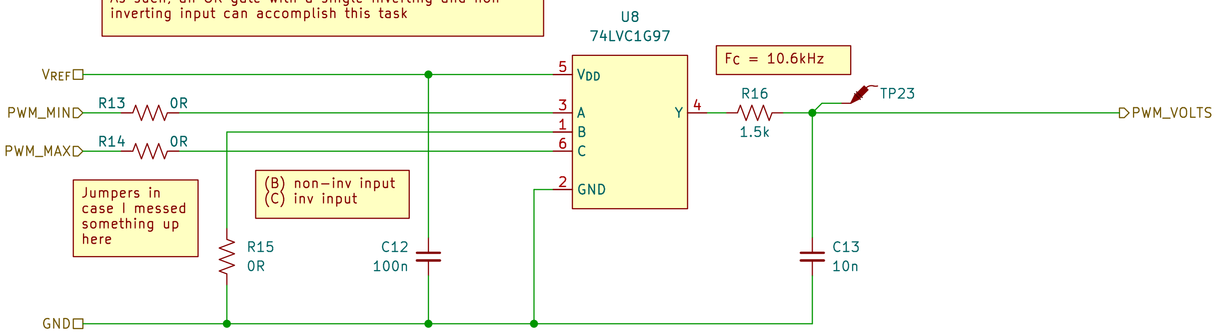


Fig. 8. **2-input NAND gate with input A inverted or 2-input OR gate with input C inverted**



Sheet: /PWM to Voltage Converter/
File: rampgen_pwm2volts.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 7.0.5		Id: 5/7

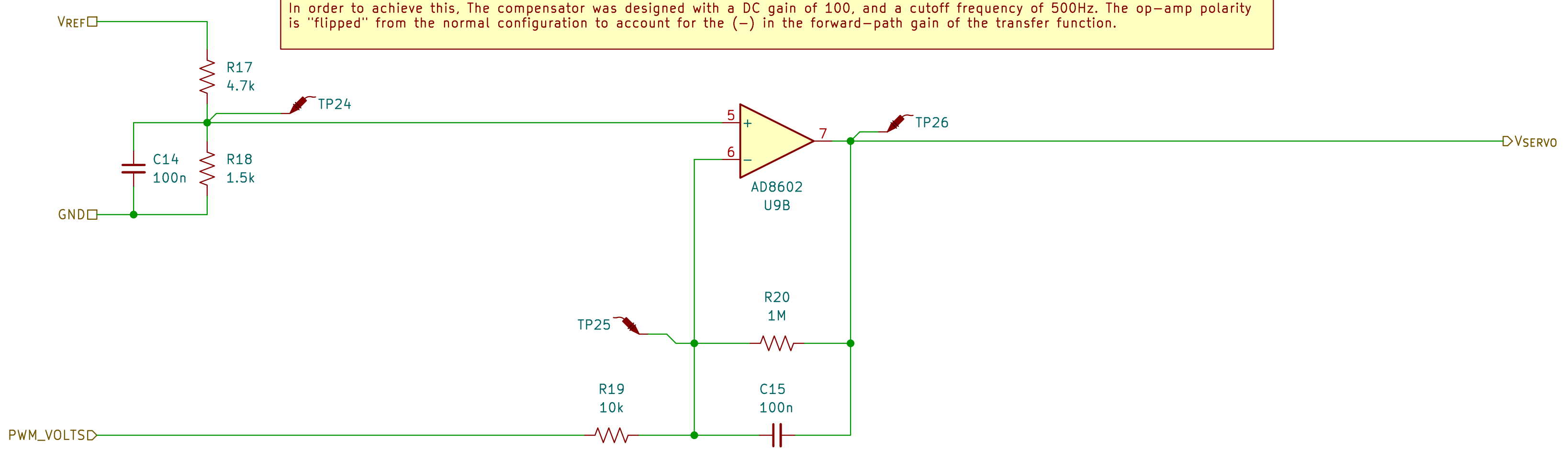
Part of this ramp generator systems includes a servo feedback loop to ensure the ramp amplitude is nearly identical with different frequency inputs and across device drift and operating conditions

The servo system works as follows:
- Sets the drive voltage amplitude of the square wave going to the integrator circuit
- Integrator circuit generates a ramp with a corresponding amplitude
- Comparators referenced to a fractional level of the reference voltage generate min/max PWM waveforms for the power stages
- The difference between the min/max PWM waveforms is generated (max – min) via boolean operations
- The average value of this difference waveform is sent back to a compensation circuit
    - If the difference is too large, ramp voltage is too low (increase drive voltage supply)
    - If the difference is too small, ramp voltage is too high (decrease drive voltage supply)

Forward path of the plant is non–linear (transfer function is proportional to $1/V_{DRIVE}$); forward path gain also varies with operating frequency (decreases at higher operating frequencies). Calculated linearized forward path gain at the desired operating point is −1.1389V/V @1.25MHz and −0.565V/V with the current operating parameters. There's a single pole in the forward path set by the frequency of the RC PWM ––> Voltage conversion filter (currently 10.6kHz).

We want the crossover frequency of the open–loop transfer function to be at most one decade below this (~1kHz) to ensure minimal overshoot of the controller step response.

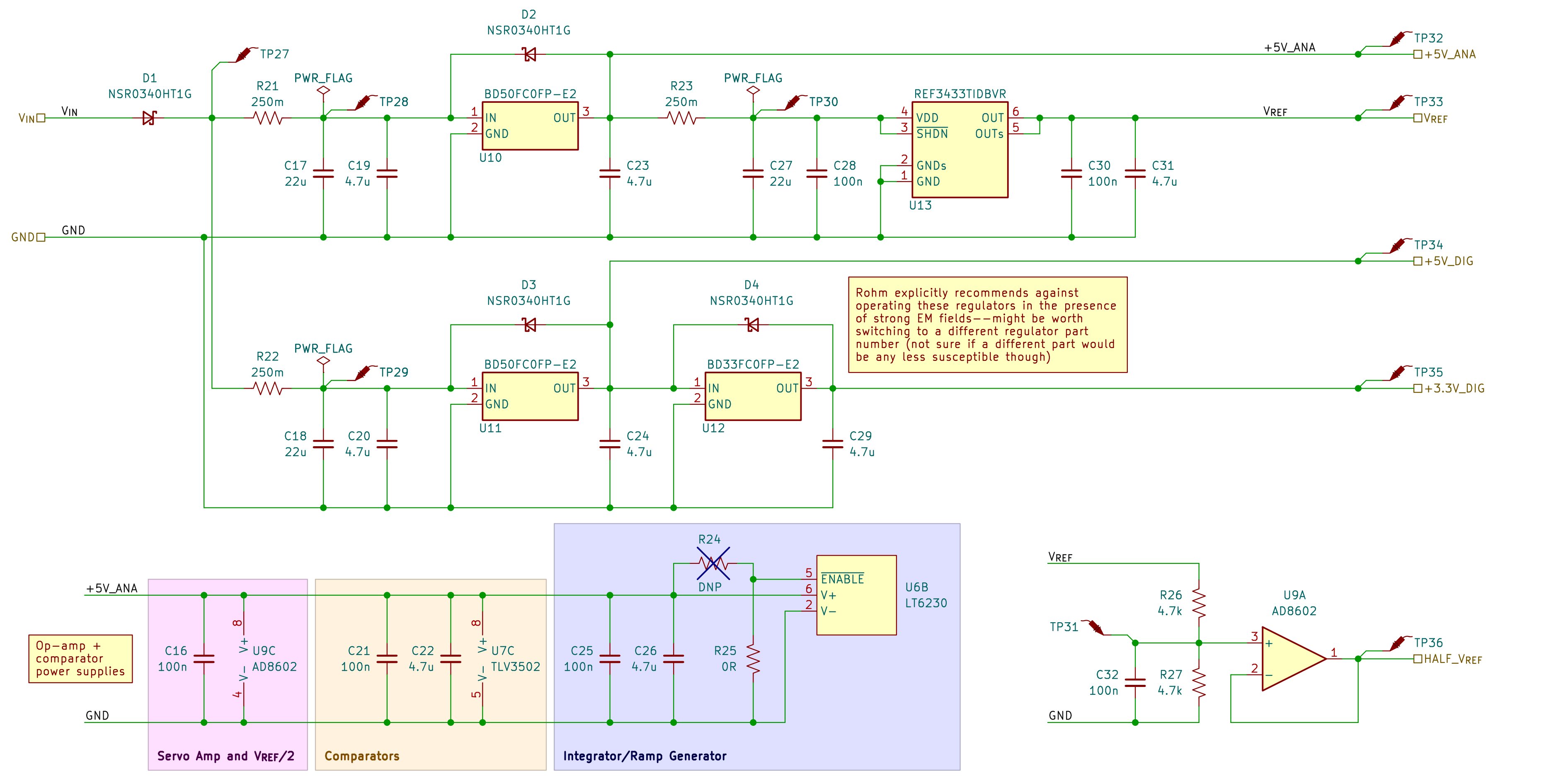
In order to achieve this, The compensator was designed with a DC gain of 100, and a cutoff frequency of 500Hz. The op–amp polarity is "flipped" from the normal configuration to account for the (–) in the forward–path gain of the transfer function.



Sheet: /Servo Amplifier/
File: rampgen_servo_amp.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 7.0.5		Id: 6/7



Regarding Power Rails:
- +5V_ANA powers ALL the op-amps in the design
  - have to be a little careful about noise introduced into the analog supply from ramp generator
  \--> specifically when amplitude modulating the driving square-wave to the integrator
- VREF operates as a reference voltage the entire design
  - ADCs
  - DACs
  - Min/Max PWM comparator references
  - Ramp servo feedback system
  - PWM to Voltage Converter for servo feedback
    \--> be careful about this
- HALF_VREF
  - Provides virtual zero-reference for all analog circuitry

In future circuit versions
- +5V_DRIVE
  - derived from input supply
  - provides 5V supplies for gate drives to the power stages
  - each channel will have its own 5V regulator for current/thermal reasons
- +5.0V Digital
  - power the fiber optic receivers - they have this awkward 5V supply voltage
- +3.3V Digital
  - power the rest of the digital circuitry
  - derive from +5.0V digital supply
  - specifically want this to match VIH and VIL thresholds of downstream analog circuitry

Sheet: /Power Supplies/
File: rampgen_supplies.kicad_sch

Title:

Size: A4 | Date: | **Rev:**
KiCad E.D.A. kicad 7.0.5 | Id: 7/7