Expertise
and insight
for the future

Scott Threet

Formula Student Can Node Design

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Electrical Engineering

Thesis

9 April 2020

Author(s) Title Number of Pages Date	Scott Threet Formula Student Can Node Design 58 pages + 4 appendices 9 April 2020
Degree	Bachelor of Engineering
Degree Programme	Electrical Engineering
Professional Major	Electronics
Instructor(s)	Heikki Valmu, Principal Lecturer

The purpose of this thesis work was development of two PCBs for Metropolia Motorsport. The first (Analog Node) reads analog sensors and sends their value on CANbus, and the second (Power Node) acts as programmable fuses and CANbus controlled switches, as well as having digital and Pulse Width Modulation (PWM) input and output capabilities.

The boards were designed using KiCad and using standard Metropolia Motorsports components. LTspice was used for circuit simulation, many calculations were done by hand, and STM32CubeIDE using the HAL libraries were used for the software development.

The result of both boards is successful, though both boards also have significant improvements that could be made in the future. For both nodes a desktop application for updating the configuration automatically could be created, or a CANbus bootloader could be implemented, but neither of those are a part of this thesis work.

There is some other software development still to do for the Analog Nodes, which will be done after the completion of this thesis. It is also possible the Analog Nodes will not perform with as high accurate resolution as initially targeted, which would involve a redesign of the board (probably using an external Analog to Digital Converter (ADC)) if higher accurate resolution than is available is required.

The improvements to the power node also would involve increased resolution, specifically of the current sense. This would probably involve a significant amount of time on software development, as it would involve a microcontroller with more pins to use as analog inputs, and the hardware would have to be significantly modified.

However, overall the PCBs are a success and will be implemented in Metropolia Motorsports newest car.

Keywords	ADC, CAN, Controller Area Network, Formula Student, Power Distribution Module

Contents

List of Abbreviations

ADC	Analog to Digital Converter.
APPS	Acceleration Pedal Position Sensors.
BMS	Battery Management System.
DLC	Data Length Code.
DMA	Direct Memory Access.
EMI	Electromagnetic interference.
HV	High Voltage.
IC	Integrated Circuit.
IMD	Insulation Monitoring Device.
IO	Input/Output.
ISR	Interrupt Service Routine.
LRC	Inductor Resistor Capacitor.
LV	Low Voltage.
NTC	Negative Temperate Coefficient.
PCB	Printed Circuit Board.
PDM	Power Distribution Module.
PWM	Pulse Width Modulation.
RMS	Root Mean Square.
RTOS	Real Time Operating System.
SWD	Serial Wire Debugging.

Glossary

I^2t	A characteristic of a fuse that can be used to determine how fast it will blow based on the current through it.
CANbus	Controller area network, a differential communication bus commonly used in automotive applications.
DCDC	A power converter that changes DC voltage to a different level of DC voltage.
DMA	Direct Memory Access, a feature of some hardware allowing it to directly write values to the memory of the microcontroller.
E12	One of the E-series of resistors defined in IEC 60063 having 12 values per decade and having 10% tolerance values.
E192	One of the E-series of resistors defined in IEC 60063 having 192 values per decade and having .5%, .25%, or .1% tolerance values.
E96	One of the E-series of resistors defined in IEC 60063 having 96 values per decade and having 1% tolerance values.
ECU	The primary control unit of the car that is responsible for taking in sensor data, and sending a torque request to the inverters.
exposed pad	An exposed metal plate on the bottom of an Integrated Circuit (IC) to reduce thermal resistance from the IC to copper.
Motec PDM15	A Power Distribution Module (PDM) manufactured by Motec with some Input/Output (IO) and CANbus functionality.
radiometric	Having an output proportional to the input, for example, a potentiometer.
ringing	Oscillation of a voltage or current that can result in overshoot.

1 Introduction

Metropolia Motorsport is a Formula Student team that builds an electric racecar every year to compete in various events around Europe. Formula Student is an international engineering competition for students. The events consist of scrutineering to ensure cars are safe and rule compliant, races, and explaining design decisions to judges. Metropolia Motorsport has been building electric cars since 2012, with the 2011 car being the last combustion car built by Metropolia Motorsport. Figure 1 shows the most recent Metropolia Motorsport car, HPF019.

Figure 1: Metropolia Motorsport 2019 car HPF019 Driving at Formula Student East

There is a need for many sensors in the car for several reasons. The car needs wheel speed and position sensors for the inverters to be able to turn the motors. The car must have Acceleration Pedal Position Sensors (APPS) (see Figure 2) so that the driver has control over the car. Formula Student rules require us to monitor brake pressure and remove power from the wheels in the case that both accelerator pedal and brake pedal are pressed at the same time. Steering angle and longitudinal and lateral acceleration sensors are used during driver training to analyze how the drivers are driving and what they need to do to improve. Similarly, suspension position sensors are used to tune the car during testing, as well as to verify the suspension design, and water temperature sensors in the cooling system are used to verify the cooling design and help plan the design for future cars.

Many of these sensors are analog sensors which need to be converted to digital format before we can do anything with the data. In the last car, the ECU read most of the analog sensors in the car, even sensors totally unrelated to the driving algorithm such as water temperature sensors. The ECU did the analog to digital conversion for these sensors and also deals with converting them from raw values to calculated values with units, and sending them to the CANbus to be logged and analyzed later by other team members.

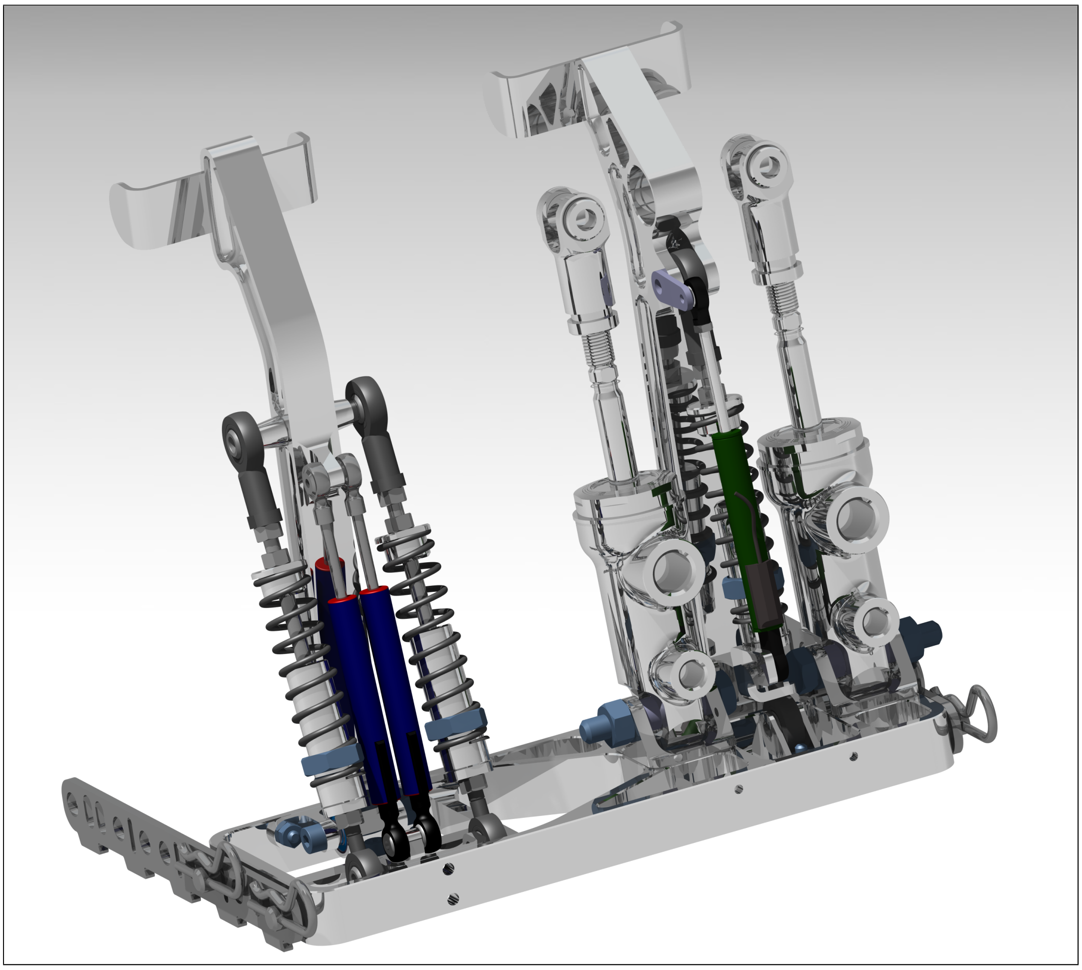


Figure 2: Rendering of the pedalbox showing APPS (in blue)

The first purpose of this project was to distribute the analog to digital conversion. This reduces the number of wires in the car, as CANbus allows for many nodes to share the same differential pair. Additionally, this project will take some of the load of converting raw values to calculated values with units off of the ECU so that the ECU can use all of its processing power for the driving algorithm, and it does not need anything related to sensors not used to drive the car. Metropolia Motorsport has recently started using a self designed ECU which has fewer analog inputs than the previous ECU, which is shown in Figure 3. A Printed Circuit Board (PCB) will be designed for this purpose which will be called the Analog Node.

Figure 3: Elektrobit 6120, the main control unit of previous Metropolia Motorsports cars

There are also many things that need power in the car. These include inverters, sensors, cooling fans, ECU, and the Analog Nodes. One of the requirements for the power is that there is some form of overcurrent protection, while the Low Voltage (LV) power supply (either a battery or DCDC) of the car has overcurrent protection, the fuse is inconvenient to change.

In the last car, Metropolia Motorsport uses Motec PDM15 (shown in Figure 4), a power distribution box with programming overcurrent limits and some digital IO and CANbus functionality. The inputs are used to monitor several digital switches in the car, and the Motec PDM15 has the ability to measure LV supply voltage and change outputs based on CANbus messages. However, using the Motec PDM15 involves many long wires to provide separate overcurrent protection to devices not near the Motec PDM15, and the

Motec PDM15 is much more expensive than the components it is made from.

Figure 4: Motec PDM15

The second purpose of this project was to provide a distributed Motec PDM15 to provide overcurrent protection for everything in the car. This device reduces the number of wires in the car and reduces the price of the car. This device also has some digital inputs which can be changed to PWM outputs, which allows digital signals such as shutdown buttons to be monitored and allows for pwm control of fan speeds. A PCB was designed for this purpose which will be called the Power Node. Finally, some of the digital IO pins are able to function as PWM input, which for example would allow us to monitor our Insulation Monitoring Device (IMD), Isometer IR155-3204 [1].

2 Project Specifications

Both nodes must be capable of sending and receiving messages over CANbus at 1 Mbit/s. Additionally, both nodes will be supplied directly from the LV supply, which has ranged from approximately 30V to 20.8V in the past. The PCBs must fit in heatshrink for waterproofing. All of the other specifications are unique to each node.

The PCBs will be used in ambient temperatures ranging from 0° C to 50° C. Both nodes must also handle rain and any humidity level. As they will be used in electric cars, interference from inverters should be considered, but considerations from combustion cars such as spark plug interference and engine heat do not exist.

As the nodes are designed to be used in Formula Student, they must have some level of reliability, but engineering students will always be available to repair them. They do not need to function perfectly, but to have sufficient accurate resolution and speed with important sensors and have as few points of failure that would stop the entire car as possible. Especially with the software it is acceptable to have bugs that do not become apparent until a specific feature is used. As long as the important features of both nodes (for example, measuring analog signals with required accurate resolution) work well, intended secondary features (for example, undervoltage detection) do not need to work.

2.1 Analog Node Specifications

Lowpass filters with a cutoff frequency much higher than we are measuring, but much lower than 10 MHz should be included on the PCB. Optimally the cutoff frequency would be about one order of magnitude higher than the frequency we are measuring, to ensure the measured signal is well above the frequency specified by Nyquist-Shannon sampling theory while still maximizing attenuation of the interference.

The target for ADC speed and resolution will be 500 Hz and .01mm resolution over a 50mm range (corresponding to suspension sensors used by Metropolia Motorsport, shown in Figure 5), which corresponds to at least 5000 steps, which requires at least a 13 bit ADC. Then conversion speed should also be fast enough that there is enough samples for a rolling average. The only requirement here that is difficult is to get higher than 12 bit resolution while using a smaller microcontroller, as most only have 12 bit ADCs integrated, and using an external ADC would take more board space.

Figure 5: Suspension Sensors on Metropolia Motorsports 2019 Car, which could require high resolution and logging frequency

The Analog Node must also be able to power sensors of various supply voltages, with some being supplied directly from LV supply voltage, some being supplied from 5V or rarely 12V. For example, from left to right Figure 6 shows a pressure sensor (being powered from 24V), an infrared temperature sensor (being powered from 5V), and a Negative Temperate Coefficient (NTC) temperature sensor (requiring another resistor and being powered from 3.3V). The microcontroller will probably require a separate supply voltage as very few modern microcontrollers run on 5V supplies.

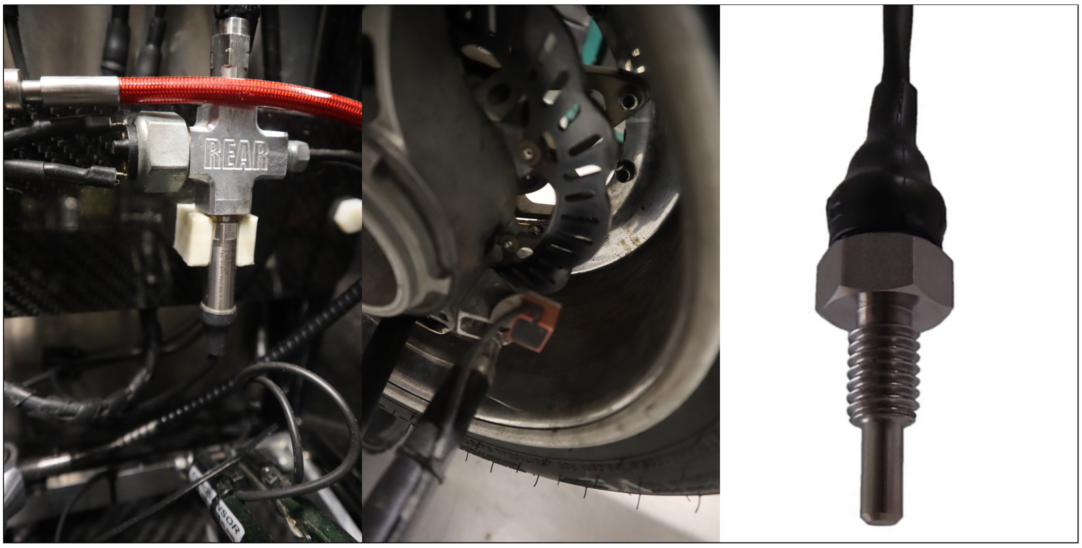


Figure 6: Various sensors used by Metropolia Motorsports requiring different supply voltages

The Analog Node must be able to either continuously output to CANbus values at a given frequency, or in response to some sort of synchronization message. The Analog Nodes must also be easy to reprogram with different sensor transfer functions or lookup tables, as well as different default settings.

The final requirements of the Analog Nodes are that they must not significantly affect the output of the sensors being read, and they must be capable of reading sensors with a wide range (24V to 5V at least) of output voltages with reasonable resolution.

2.2 Power Node Specifications

The ADC resolution requirement for the Power Node is lower than for the Analog Node. The target for conversion speed will be determined so that the Power Node can switch off before the main fuse blows and prevent it from blowing due to a short circuit, however, the LV main fuse is not guaranteed to be the same and is just some 10A automotive fuse, for example the fuses used last year are shown in Figure 7.

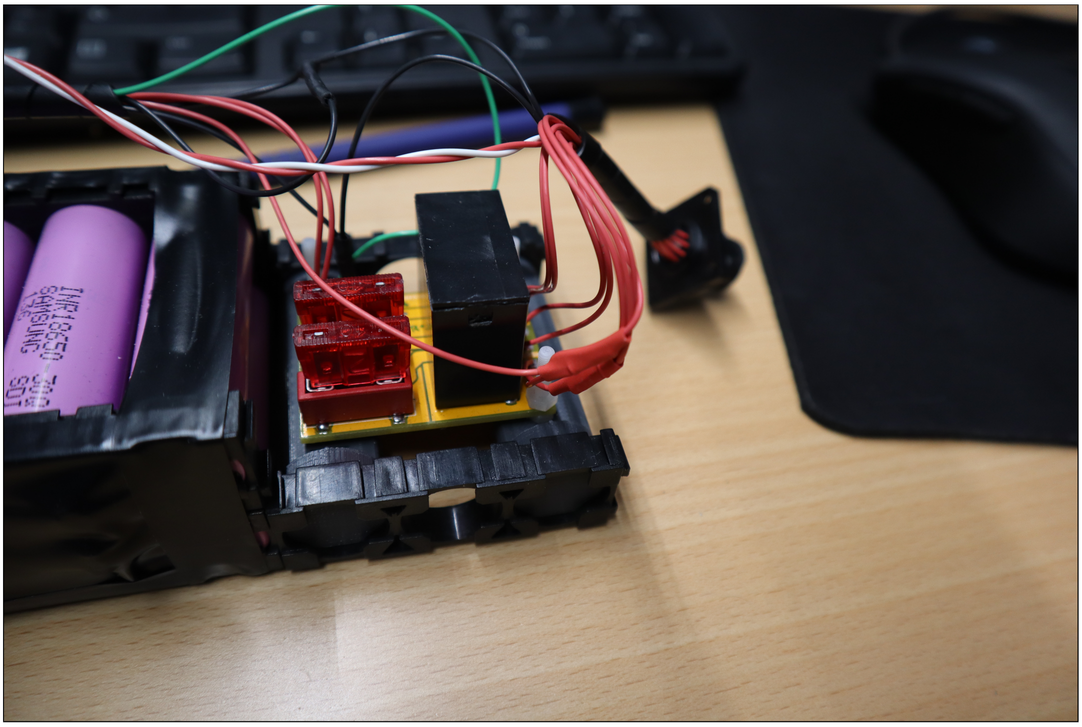


Figure 7: Metropolia Motorsport HPF019 LV main fuses.

Additionally, a high side driver is required and the high side driver must have its own integrated overcurrent shutoff, so the software only needs to react faster than the fuse in situations where the high side driver wouldn't react. The chosen high side driver must have current monitoring integrated. The Power Node must be capable of driving at least a 10A total load or 4A on one channel and switching all reasonable values of the LV supply voltage.

The only required regulated voltage for the Power Node is whatever is required to supply its own microcontroller and provide a reasonable PWM signal for control of main cooling fans (for accumulator, inverters, and motors).

Finally, the software must be able to switch outputs and change PWM duty cycles based on CANbus messages as well as send digital input states over CANbus. The Power Node must be relatively easy to change overcurrent limits in software and reprogram with different settings.

3 Component Selection

3.1 Microcontroller and ADC

STM32 was chosen as the manufacturer, as Metropolia Motorsport uses STM32 microcontrollers in other places and using all microcontrollers in the same family makes the programming easier. The smallest STM32 that can be hand soldered was selected that also had the required features.

The smallest STM32s (ignoring BGA, CSP, and UFQFPN) are SOIC-8, TSSOP-14, TSSOP-20, and LQFP-32 (shown in Figure 8). The choices were narrowed down further as only the TSSOP-20 and LQFP-32 packages offer CANbus functionality, so the microcontrollers to choose between were STM32F042Fx, STM32F334Kx, STM32F303Kx, and STM32G441Kx. As they are selected primarily for the characteristics of their ADCs, these are compared in Table 1.

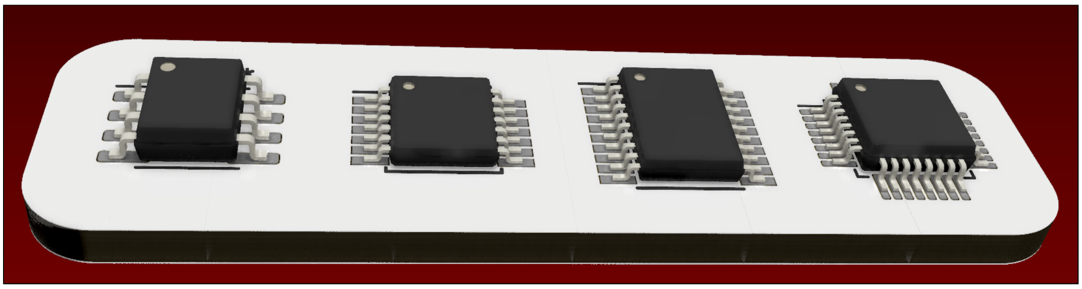


Figure 8: Comparison of SOIC-8, TSSOP-14, TSSOP-20, and LQFP-32 packages.

Table 1: Comparison of STM32 microcontroller ADCs.

	F042	F334	F303	G441
ADCs	1	2	4	2
samples/second	1M	5M	5M	4M
resolution	12-bit	12-bit	12-bit	12-bit (16-bit with hardware oversampling)

While all the conversion speeds are fast enough, the 12-bit resolution is not good enough for the Analog Node. The hardware oversampling requires 4 measurements at a lower bit count in order to increase the bit count by 1. Therefore to use the ADC as a 13-bit ADC, 4 12-bit samples must be taken for each 13-bit sample, meaning the max speed is only 1M samples/second for STM32G441Kx. However, as using the ADC as a 16-bit ADC there is still 15.625k samples/second, which would still allow a 30 sample rolling average to be taken in order to have a logging frequency of 500Hz if no other channels are used on that Analog Node, so the STM32G441Kx microcontroller is chosen for the Analog Node. [2.]

For the Power Node, the only requirement other than having an integrated ADC was that the chosen microcontroller must have sufficient IO and especially timers for PWM generation. STM32G441Kx has plenty of timers, so it was chosen to be used for the Power Node as well [2].

3.2 High Side Drivers

As the high side driver for the Power Node must carry more current, that one was chosen first.

3.2.1 Power Node High Side Drivers

The target for the power node was to have up to six power outputs, so drivers with 2 and 6 channels were looked at (none were available with 3 channels). Several product families from different manufacturers are available, with Infineon's PROFET series, TI's smart power switch series, and STM32's high side switch series [3;4;5]. The highest current devices in the LV system take approximately 1.5A, but that will increase as we need more powerful fans and pumps for cooling.

The smallest PROFET available was an SOIC-8, but it is only rated to drive up to 4A. The next smallest package with current monitoring and sufficient voltage rating was a 5-pin DPAK, which comes with rated nominal currents of 10A and 16mΩ resistance, so this component would be sufficient for our uses. There was also the option of using a 20-pin TSDSO package, which has an exposed pad that is difficult to solder, but two channels each rated for a nominal 5.5A and 30mΩ resistance. This would be preferred for the Power Node, as it takes less space per output than the single channel PROFET and is still sufficient for our requirements. [3.]

TI has several 1, 2, and 4-channel drivers, in their TPS series of high side drivers. However, all of their high-side drivers have open load detection by undercurrent, which is fine when we are supplying devices that draw significant current, but there are several devices in the car that take so low current that open load may be detected incorrectly, so the TI high side drivers are not chosen. [4.]

The STM32 high side drivers only come in two packages, a single SOIC-8 option and Power SSO-16. The SOIC-8 component has high enough resistance (140mΩ) that it is likely to overheat, but the Power SSO-16 has options for lower resistances, including a 2-channel chip with 20mΩ. The Power SSO-16 also has an exposed pad, so will be difficult to hand solder. However, the other feature of STM32 high side drivers is that some have Multisense, an option of multiplexing the current monitoring between voltage and temperature measurements as well. [5.]

The Multisense enabled 2-channel 22mΩ VND7020AJ high side driver from STM32 was chosen to be used for the Power Nodes [6]. All of these drivers are also designed to capable of driving inductive and capacitive loads without being damaged. [5.]

3.2.2 Analog Node High Side Driver

The Analog Node high side driver only needs to drive current enough to supply sensors which will draw at most about 20mA. The high side driver is only used for sensors supplied directly from the LV power supply. As any chosen DCDCs include overcurrent protection, the only thing to be gained from a high side driver is current monitoring, which is not a common feature in low voltage high side switches.

The SOIC-8 STM32 1-channel 140mΩ VN7140ASTR was chosen for the Analog Node as it can switch sufficient voltage and provides some current monitoring. The package is smaller and easier to solder than the one chosen for the Power Node, and due to not powering anything that takes more than 20mA, overtemperature is not a concern for the Analog Node high side driver. [[7].]

3.3 Can Transceiver

As both nodes have a 3.3V supply, SN65HVD230 was chosen as the CANbus transceiver [8]. This is the standard can transceiver for Metropolia Motorsport and has been used in several PCBs in the car already.

3.4 Oscillator

STM32G441Kx has an internal oscillator that can work for CANbus in practice, but the recommended tolerance for CANbus is .49% tolerance if a prescaler of 1 is used, which is much lower than the tolerance of the internal oscillator [9]. Therefore, an external oscillator is used for improved reliability. An oscillator is used instead of a crystal for simplicity.

The CANbus controllers work best when clocked with a multiple of 16 MHz, so a 16 MHz oscillator was chosen. ECS-3225MV-160-BN was chosen for its small package and because it is part of a product family that offers 32MHz and 48MHz options in the same package [10]. Additionally, this oscillator was in stock at the time of the board design, in the case that EC-3225MV-160-xx is not in stock, and 32/48MHz options in the same family are not in stock another oscillator can be used as long as it has compatible dimensions. ECS-3225MV-160-BN has a frequency stability of 50ppm, which is easily sufficient for the CANbus in the car.

3.5 Operational Amplifiers

A single-supply or rail-to-rail operation amplifier is required, as we do not have a negative supply voltage available and putting an inverting DCDC on the board would require a significant amount of board space. The unity gain bandwidth is not an issue, as the frequencies we are interested in measuring are so low. However, high supply and input voltages are required, as some sensors could have an output voltage up to the LV supply voltage, which could be up to 30V.

LM358 is the standard operation amplifier used by Metropolia Motorsport. LM358 is a dual single-supply operational amplifier with absolute maximum input voltages rated independently from the supply voltages, so it does not break if the input voltage is higher than the supply voltage. As it is a general part and made by different manufacturers, the specifications for LM358 vary. The STM32 LM358 is used as Metropolia Motorsport is able to get some parts as samples from STM32. This operational amplifier has an input offset voltage of up to 9mA, which can be nulled in software. However, it has a input offset voltage drift of up to 30uV/°K, which means if the offset voltage is nulled at 10°C, then at 40°C the input offset voltage could have changed by 900uV, which means LM358 is too inaccurate to measure 5000 steps in a signal ranging from 0V to 3.3V. [11.]

Due to the high input offset voltage of LM358, an alternative operation amplifier was selected. LM358 is good for most purposes, but a separate operation amplifier must be used when a 5000 step resolution is wanted. Fortunately, ADA4522-2 is a rail-to-rail zero-drift operation amplifier that is pin-compatible with LM358. Further, it has supply voltage that ranges from 4.5V to 55V. ADA4522-2 also only has an input offset voltage at most of 7.2uV, so it is not even necessary to null the input offset in software. [12.]

3.6 DCDC Converters

A switching regulator is used on both the Analog Node and Power Node for improved efficiency. The Power Node only requires 3.3V, so the switching regulator will supply that directly. However, for the Analog Node we would like a better quality power supply at least for the ADC. Additionally, the Analog Node must have a voltage that can be 5V, and should have the option for at least 12V. Therefore, a switching regulator will be used on the Analog Node to supply sensors, and a linear regulator will drop the voltage from that to 3.3V.

3.6.1 Buck Converter

Metropolia Motorsport uses a circuit with LT3970 for buck converters that do not need to output more than 535mA. Therefore, for the switching regulators this circuit and component is used. Voltage dividers for 3.3V and 5V output are already calculated and Metropolia Motorsport has precise resistors for the feedback in stock. Because both of those voltage dividers use 1MΩ high side resistors, 12V output was calculated with the same high side resistor. As the feedback voltage for LT3970 is 1.21V, the low side resistor closest with standard values is 112kΩ either in the E96 or E192 series [13]. [14.]

The maximum output of the buck converter is required to determine the power dissipation requirement for the linear regulator. 20V was used for this calculation, as any sensor requiring more than 20V will be supplied from the LV supply voltage.

3.6.2 Linear Regulator

The chosen linear regulator was chosen to supply the microcontroller, can transceiver, oscillator, and possibly external sensors (especially potentiometers and NTC resistive temperature sensors). Ideally separate regulators would be used for the can transceiver, oscillator, and digital supply for the microcontroller, but these consume low current and adding another linear regulator would take a significant amount of board space, so only a single linear regulator is used. The most current that STM32G441Kx can draw is 34mA, SN65HVD230 can draw up to 17 mA, and ECS-3225MV-160-BN has up to 4mA supply current, making a requirement of 55mA output power [2] [8] [10]. As external sensors should not take more than 1mA each, the target current is 59mA output current while the switching regulator is at the highest voltage level that it may need to be. With the 20V decided previously, this corresponds to 1.02W of power that must be dissipated by the linear regulator at most.

Other than power capabilities and size, tolerance and ripple rejection were considered in selection. Especially ripple rejection was considered carefully, as manufacturers do not always give ripple rejection at the same frequency, so a higher value may still be worse if the ripple rejection is given at a much lower frequency. Especially important are the frequencies around 600kHz (the switching frequency of LT3970).

SPX1117-L-3-3 was chosen as the best option in a SOT-223 package. This linear regulator has a tolerance of 1%, line regulation of .21%, load regulation of .36%, and more importantly ripple rejections that is at worst about 27 dB and more than 50 dB at 600kHz. Additionally, SPX1117-L-3-3 has a rated long term stability of .03% and voltage changing at most by .44% with temperature (with output loads from 10mA to 800mA), so if a specific Analog Node is calibrated to specific sensors we have a higher resolution. Additionally, SPX1117-L-3-3 has overcurrent and overtemperature protection, at least one of which is required. [15.]

3.7 Power Node Fuse

The Power Nodes must have a fuse for overcurrent protection of the traces on the Power Nodes. The fuse was chosen to have at least 30V DC rating, a maximum interrupt current of at least 630A corresponding to a theoretical short circuited lithium ion battery discussed in Wiring and Supplied Device Overcurrent Protection, and an I^2t rating of less than .5A^2s as calculated in Wiring and Power Node Fusing. Bel Fuse 0678H0500-02 was chosen for this purpose, having a max DC voltage of 125V, a maximum interrupt current of 2kA DC, and an I^2t rating of .03 A^2s, and a breaking current of 500mA [16].

3.8 Other Components

0805 package (imperial sizing) resistors, capacitors, inductors, and ferrite beads are used as they are easy to hand solder and commonly used by Metropolia Motorsport. Additionally, capacitors and inductors are commonly available with sufficient capacitances, voltage ratings, inductances, and current ratings. SOD-323 package zener diodes and SOD-123 package diodes are used for the same reasons. All chosen inductors are shielded to reduce potential Electromagnetic interference (EMI) issues.

Serial Wire Debugging (SWD) pinheader was chosen as a 2.54mm pinheader both as it is commonly used for programming connections and it allows us to easily solder the 22 AWG wires that are used by Metropolia Motorsport, which can be used to make much more reliable programming connectors than the pinheader, as shown in Figure 9.

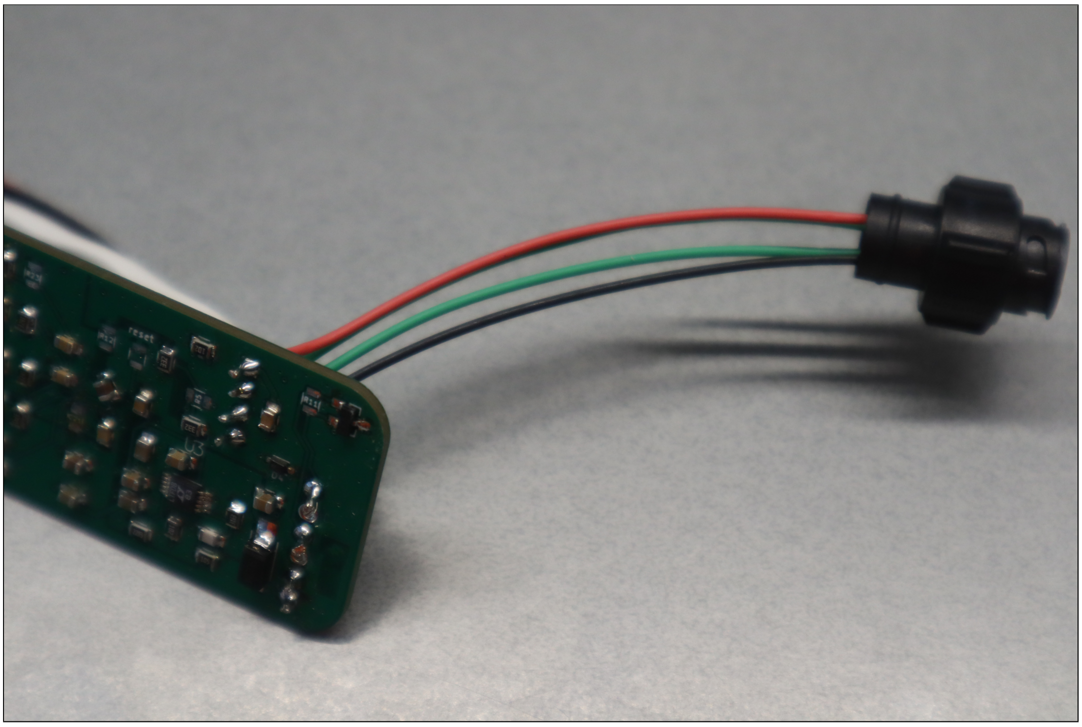


Figure 9: Wires soldered to a board for use of a wire to wire connector for SWD programming.

E12 (10% tolerance) resistors are used, as Metropolia Motorsport has a set of E12 resistors from 1kΩ to 820kΩ. For analog inputs, .1% resistors are used as a trade off of price against accuracy. The resistor tolerance is not especially important as it is very easy to calibrate the error from them in software. More importantly, those resistors were chosen to have a relatively low thermal coefficient of resistivity of 10ppm.

4 Theory and Calculations

4.1 Lowpass Filter Design

Lowpass filters were designed primarily to reduce interference on the Analog Node measurements. An Inductor Resistor Capacitor (LRC) filter was initially chosen to have the attenuation of a second order filter without any active filtering. This circuit is used on both sides of a buffer. The buffer is used for high input impedance (to reduce the effect of the measuring device on the measurement). On the output of the buffer the filter is built into the voltage divider so that the resistor in the voltage divider prevents ringing, while the ringing on the output of the buffer is prevented by the high side resistor of the voltage divider (see Figure 10). On the input of the buffer, a series resistor is included to prevent ringing. Values were chosen so that the cutoff frequency is well over 500Hz (at least 1kHz to satisfy the Nyquist-Shannon sampling theorem) and well under 1MHz. The interference frequency is not specifically known, but in the 2018 and 2019 cars the interference from motors and inverters was somewhere from 10MHz to 20MHz.

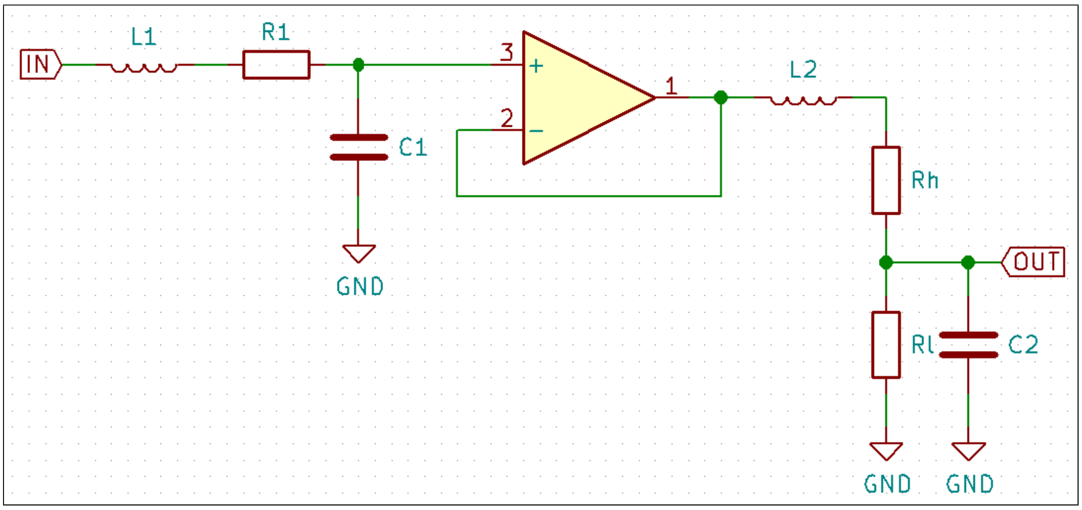


Figure 10: Filter Schematic

Two second order lowpass filters as shown in Figure 10 are used. Shielded inductors were chosen to reduce the potential interference, as high enough inductances are easily available in shielded inductors of the used package. The cutoff frequency was determined by an AC sweep simulation as shown in Figure 11. The circuit was checked for ringing by a simulating it with a step response input as shown in Figure 12, with ringing circled in red.

Note that the first lowpass filter (L1, R1, C1) is affected by the output impedance of the used sensor. If a sensor has high enough output impedance to cause the cutoff frequency to drop to be near the desired measurement frequency, this should be considered, for example, by reducing the size of R1 or shorting L1. For example, this would be the case when a linear potentiometer (for example, 25kΩ) is used and high frequency (for example, 500Hz) is desired.

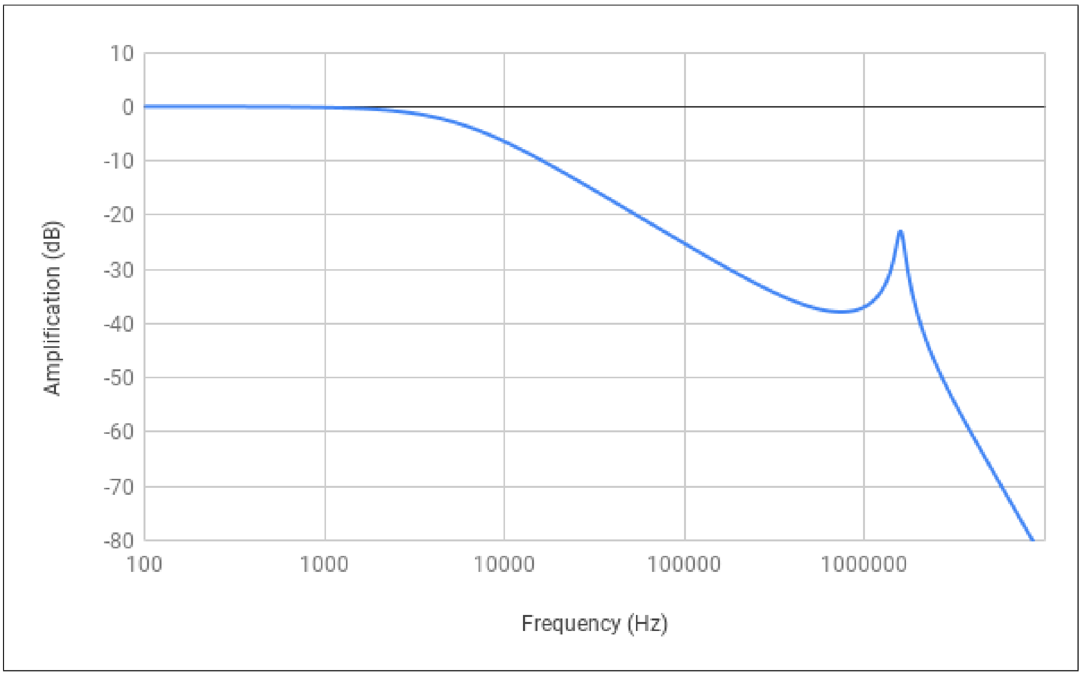


Figure 11: Simulation result of the lowpass filter in frequency domain.

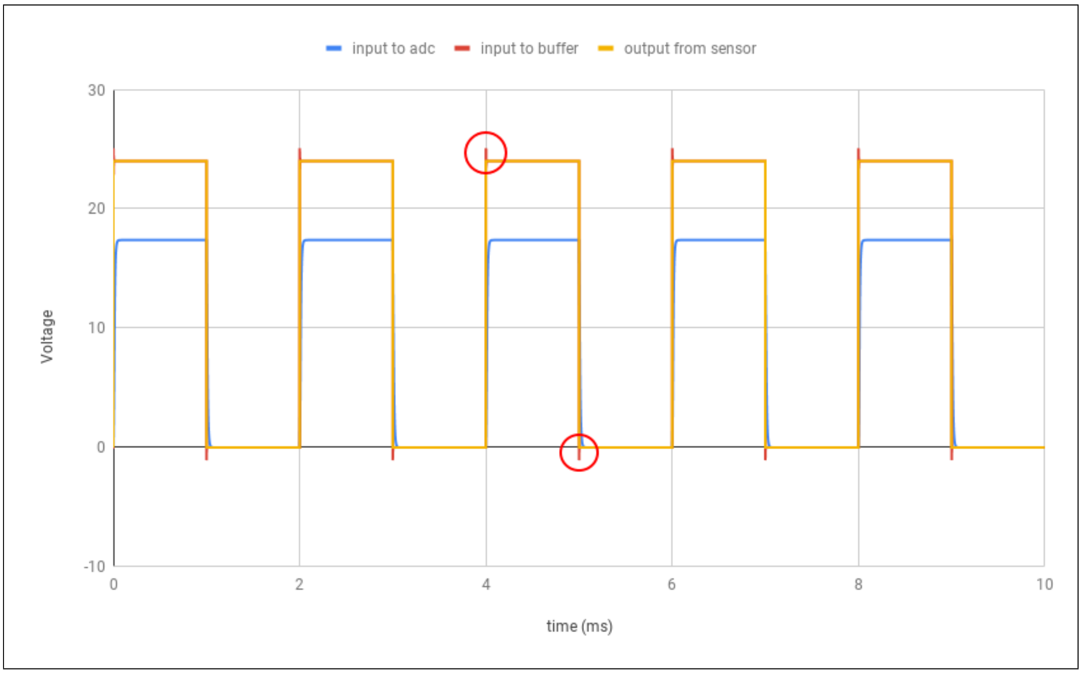


Figure 12: Simulation result of the lowpass filter with a step response (ringing on the input to buffer circled in red).

The inductor was chosen to be 1uH and the capacitor was chosen to be 1nF, giving a cutoff frequency of approximately 130kHz at most and approximately 5kHz when the high side resistor is 26.7kΩ. Still, moving the cutoff frequency lower to have more attenuation at higher frequencies when the high side resistor is smaller is desirable, so for the more common case if the high side resistor is 1.21kΩ or 1.74kΩ, 10nF is preferred, bringing the cutoff frequency down to about 13kHz or 9kHz, but for higher resistances lower capacitance should be used to avoid ringing and too low a cutoff frequency.

4.2 Analog Node ADC Error Calculations

The analog reading has several sources of error to be considered. The first of these is the accuracy of the sensor, which is outside the scope of this project, though the sensor accuracy and resolution should be considered when selecting sensors if the user has a target sensors resolution. Also outside the scope of these calculations is the error from sensor output impedance, though sensor output impedance was considered in the design. This is the reason that buffers are used. There is also some interference to consider on both the signal and the power supply, which was considered in the design with lowpass filters. However, they are not calculated as the interference levels are unknown and change depending on what motors and inverters are being used. Similarly, the effects of ground voltage offset are considered in the design by having an analog and digital ground tied together under the ADC power supply pins on the microcontroller, but this source of error was not calculated.

There is also noise from the resistors in the voltage dividers and operational amplifiers to be considered, though noise is not a major source of error. The tolerance in voltage divider resistors and operational amplifier input offset voltage (and current) are sources of error as well. Both the resistors and input offset voltage drift with temperature, so while some of the sources of error can be nulled in software, there will be some error as the temperature changes. The power supply tolerance and ripple directly affect the ADC accuracy as well, though the tolerance can be nulled in software, the output voltage will also drift with temperature. The final error that was considered is the quantization error that comes from converting an analog value to a digital value.

The error was calculated for two use cases. The first use case was using LM358 and not calibrating it. In this case the worst voltage division option (with respect to accuracy) was be taken into account. For the second use case, ADA4522-2 was considered and software calibration was considered mandatory. Additionally, the second use case was only be considered with no voltage divider and a radiometric sensor.

4.2.1 Noise

For both use cases, the noise from the sensors was not considered, as the specific sensor is unknown. For the second use case especially there should be sufficient white noise for the hardware oversampling, so having white noise dominant and of at least 1 bit amplitude (806uV with 3.3V range, 12 bit) was desired for the second case. A rolling average will be used in both cases, but not calculated into the accuracy. The switches were considered ideal, so the noise was calculated the same as if they were not there, as when an ideal switch opens it only freezes the instantaneous value, but does not affect the noise level otherwise. For the first case, the circuit considered for the noise for is shown in Figure 13.

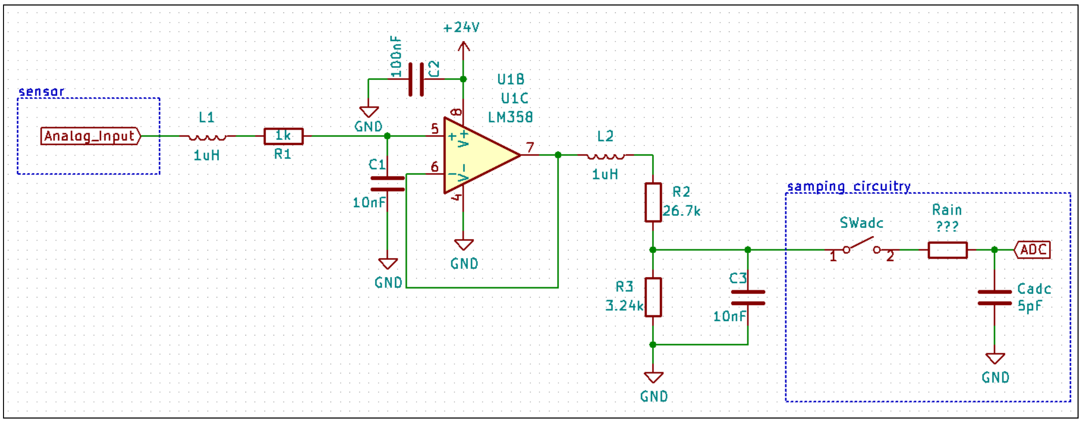


Figure 13: First use case schematic.

Unfortunately, the value of R_{ain} (internal analog resistance as seen in Figure 13) is not given in the datasheet for the microcontroller, so R_{ain} can not be used to calculate the noise [17]. Due to this the noise at the input of the ADC will be calculated, ignoring the 5pF capacitor as it will have very little effect on the bandwidth given the 10nF capacitor that is also used.

For this case, the worst case (R2=26.7k) noise magnitude was first calculated. For the purposes of this, first the bandwidths of noise at resistors R1, R2, and R3 were determined by simulation replacing the operational amplifier with a voltage controlled voltage source (an ideal voltage follower). In order, the approximate bandwidths are 5.0kHz, 5.5kHz, and 5.5kHz. In the case of R1 the operational amplifier was not factored into the bandwidth as just the bandwidth of R1 and C1 is already much lower than the unity gain bandwidth of the operational amplifier.

Thermal noise in resistors ($V_{thermalnoise}$) is given as a function of the Boltzman constant ($k = 1.38 * 10^-23$), the temperature (T), the resistance (R), and the bandwidth (Δf). Equation 1 is given for magnitude of the noise and not the usual situation of the Root Mean Square (RMS) of the noise.

$$V_{thermalnoise} = 2\sqrt{2kTR\Delta f} \quad [V] \tag{1}$$

As the highest temperature the nodes are expected to be at is 50°C, 323.15°K was used as the temperature to calculate thermal noise for this situation. The calculated approximate thermal noise magnitudes are 422nV, 2.29uV, and 797nV. The total noise from the resistors was then approximately 3.51uV.

While there are many different sources of noise in operational amplifiers, the LM358 operational amplifiers are only rated with the input noise voltage, which is 55nV per root Hertz [18; [11]]. A simulation was again used to determine the bandwidth of that noise as 5.5kHz (the same as R2 and R3), which gives a noise voltage magnitude of 4.08uV. This brings the total noise for this use case to 7.59uV, which is only slightly lower than one significant bit.

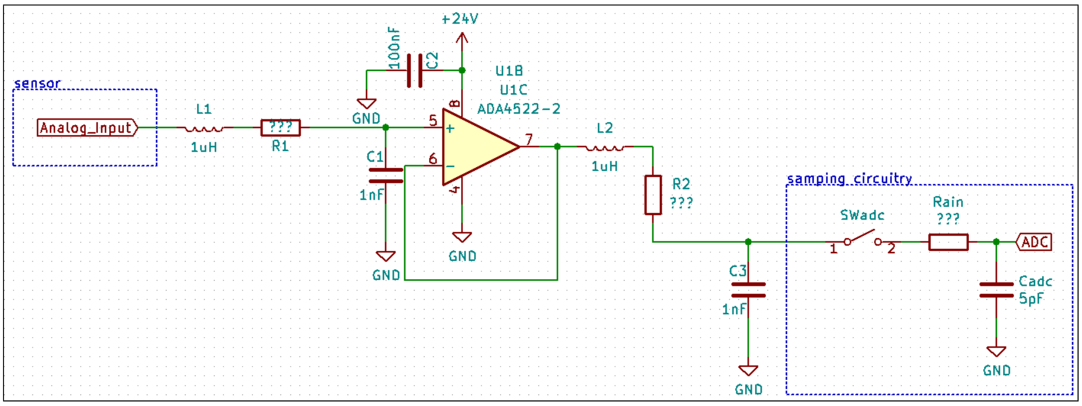


Figure 14: Second use case schematic.

For the second case, the circuit that we calculate accuracy and noise for was slightly different and is shown in Figure 14.

For the second use case, the noise, especially from R2 is more important, as there should be enough noise or equivalent ripple at the input of the ADC for the hardware oversampling to work correctly. The easiest way to do this is to rely on interference to produce a white noise equivalent signal, but unfortunately this cannot be relied upon as most of the sources of interference occur periodically and are at set frequencies. The interference of the system may be sufficient for oversampling to work, but given the spectrum of the interference is unknown this is unreliable.

As R_{ain} is not specified in the microcontroller datasheet, it is incredibly difficult to do any reasonable calculation here, as large values of R_{ain} will lead to a lowpass filter with the sampling capacitor that could greatly reduce the noise. Because of this, in order to achieve greater than 12 bit resolution, testing must be carried out.

Therefore, hardware oversampling will be tested by using large values of R1 and possibly removing C1 and C3 from the circuit. If the bandwidth is too high, the noise from ADA5422-2 is no longer white, but the noise from ADA5422-2 is not significant because the noise will be dominated by R1 [12].

4.2.2 Resistor Tolerances

Voltage Divider tolerances were considered for the first case, for when the analog input is not calibrated in software. The voltage divider tolerances were only considered for the DC situation, as the cutoff frequency has a wide acceptable frequency range. The ADC resistance to ground is not given, so this resistance was not considered in the calculation. Similarly, while the operational amplifier input impedance is not given, it should be large enough to disregarded. Both of these will form parasitic voltage dividers, but the ADC input is rated to have some input impedance, and operational amplifiers have an input impedance that should be several orders of magnitude higher than the output impedance of any sensor used by Metropolia Motorsport.

To calculate the error $(E+/E-)$ from the voltage dividers, the maximum $(V_{out(max)})$, ideal$(V_{out(ideal)})$, and minimum$(V_{out(min)})$ voltage outputs of a voltage dividers were calculated from the tolerance $(t$, the high side resistance corresponding to R2 (R_h), and the low side resistance corresponding to R3 (R_l).

$$V_{out(ideal)} = \frac{V_{out}}{R_l + R_h} R_l \quad [V] \tag{2}$$

$$V_{out(max)} = \frac{V_{out}}{R_l(1+t) + R_h(1-t)} R_l(1+t) \quad [V] \tag{3}$$

$$V_{out(min)} = \frac{V_{out}}{R_l(1-t) + R_h(1+t)} R_l(1-t) \quad [V] \tag{4}$$

From Equation 2, Equation 3, and Equation 4, the error from the voltage divider was calculated.

$$E+ = \frac{V_{out(max)}}{V_{out(ideal)}} - 1 = \frac{2tR_h}{R_h(1-t) + R_l(1+t)} \tag{5}$$

$$E- = 1 - \frac{V_{out(min)}}{V_{out(ideal)}} = \frac{2tR_h}{R_h(1+t) + R_l(1-t)} \tag{6}$$

For the second use case, even if the analog input was not calibrated in software, resistor tolerances would not be considered, as there is not actually any voltage divider that could cause an error. A large input resistance would form a parasitic voltage divider with the approximately 100GΩ common mode resistance of the operational amplifier, but it will be insignificant, unless a sensor with very high output impedance is used [12].

4.2.3 Resistor Temperature Coefficient

The cause for most of the changes in component specifications for a specific component come from temperature, so a temperature range for the used sensors must be determined. While Metropolia Motorsport sometimes drives in freezing weather, the testing is done in the summer and the data from such a cold track is not considered useful, so 0°C was considered the minimum temperature. However, the competitions are in the summer usually in places that are fairly warm, and the Analog Nodes may be warmer than average even though they should not generate significant heat, so 50°C was considered as the maximum use temperature. As the calibration is done inside the temperature will range from 15°C to 25°C, meaning the range for temperature drift is +35°C/-25°C.

Starting with the previous equations for voltage divider output maximum and minimum voltages, the change due to the temperature coefficient (α). These calculations are calculated as if the rated temperature coefficient is symmetrical.

$$V_{out(max)} = \frac{V_{out}}{R_l(1+t)(1+35\alpha) + R_h(1-t)(1-35\alpha)} R_l(1+t)(1+35\alpha) \quad [V] \tag{7}$$

$$V_{out(min)} = \frac{V_{out}}{R_l(1-t)(1-35\alpha) + R_h(1+t)(1+35\alpha)} R_l(1-t)(1-35\alpha) \quad [V] \tag{8}$$

From Equation 2, Equation 7, and Equation 8, the error was again calculated.

$$\begin{aligned} E+ &= \frac{V_{out(max)}}{V_{out(ideal)}} - 1 - E+_{tolerance} \\ &= \frac{(2t+70\alpha)R_h}{R_h(1-t)(1-35\alpha) + R_l(1+t)(1+35\alpha)} - \frac{2tR_h}{R_h(1-t) + R_l(1+t)} \end{aligned} \tag{9}$$

$$\begin{aligned} E- &= 1 - \frac{V_{out(min)}}{V_{out(ideal)}} - E+_{tolerance} \\ &= \frac{(2t+70\alpha)R_h}{R_h(1+t)(1+35\alpha) + R_l(1-t)(1-35\alpha)} - \frac{2tR_h}{R_h(1+t) + R_l(1-t)} \end{aligned} \tag{10}$$

Equation 11 and Equation 12 are easier to use to approximate the maximum allowed temperature coefficient for a required resolution. With allowed error of .025% ($\frac{1}{2^{12}}$ corresponding to 12 bit resolution), this means that the highest allowable temperature coefficient with our worst case voltage divider of 26.7kΩ and 3.24kΩ is 4ppm, which is very low and not reasonable to purchase. Even the most common voltage divider of 1.74kΩ and 3.24kΩ requires 7ppm with the same target resolution, which is still not reasonable.

$$\alpha <= \frac{E(R_h + R_l)}{\Delta T(2R_h E(R_h + R_l))} \tag{11}$$

$$\Delta T <= \frac{E(R_h + R_l)}{\alpha(2R_h E(R_h + R_l))} \tag{12}$$

If higher resolution is required, the analog inputs must be calibrated at some temperature and then used at a similar temperature, or low temperature coefficient resistors must be purchased. Still, 10ppm resistors are purchased initially, which will cause an error at most .07%.

The error from the resistor temperature coefficients is not considered for the second use case, as there are no intended voltage dividers.

4.2.4 Input Offset Voltage

The input offset voltage causes a voltage error directly on the output of the operational amplifier. The input offset voltage error is attenuated by the voltage divider. For the LM358 operational amplifier (first use case) this is at most 9mV [11]. The voltage error from input offset voltage (V_{io}) that actually is found at the ADC input is then calculated.

$$V_{io} = \frac{.009}{R_l * R_h} R_l \quad [V] \tag{13}$$

The maximum input offset voltage was used for calculating the total error and determining the worst case voltage divider.

For the second use case the input offset voltage was not considered as it will be nulled in software.

4.2.5 Input Offset Voltage Drift

The input offset voltage drifts with the temperature. For the first use case, the error from the input offset voltage was considered as the maximum input offset voltage. However, if LM358 is used and the analog input is calibrated in software, the input offset voltage error to be considered is at most 30uV/°K, which is 1.05mV input offset voltage over the previously defined temperature range [11].

For the second use case the input offset voltage is the first noticeable source of error, as no voltage divider is used to cause errors and input offset voltage can be nulled. The ADA4522-2 operational amplifier has an input offset voltage drift is at most 15nV/°K, corresponding to 525nV over the previously defined temperature range [12].

4.2.6 Worst Case Voltage Divider

To check what the worst case voltage divider is for the first use case, the error from input offset voltage and from the resistor tolerances were compared.

$$V_{io(max)} = \frac{.009}{R_l(1+t) + R_h(1-t)} R_l(1+t) \quad [V] \tag{14}$$

$$E = \frac{(2t + 70\alpha)R_h}{R_h(1t)(135\alpha) + R_l(1t)(135\alpha)} - \frac{2tR_h}{R_h(1t) + R_l(1t)} \tag{15}$$

As the low side resistor is 3.24kΩ, the temperature coefficient of the resistors is at most 10ppm, the tolerances are at most .1%. To convert the percentage error from the voltage divider to voltage, the following equation is used.

$$V_{e(resistor)} = V_{measured} * E \quad [V] \tag{16}$$

3.3V was considered as the maximum measured voltage to calculate the maximum error, as using a lower value as the maximum value would result in a lower resolution. This total error of operational amplifier and voltage divider is the sum of these errors.

$$\begin{aligned} V_{e(total)} = {} & \frac{.009}{R_l(1+t) + R_h(1-t)} R_l(1+t) \\ & + 3.3\left(\frac{(2t + 70\alpha)R_h}{R_h(1t)(135\alpha) + R_l(1t)(135\alpha)} - \frac{2tR_h}{R_h(1t) + R_l(1t)}\right) \quad [V] \end{aligned} \tag{17}$$

As all the values except the high side resistor are set, values were substituted to check with what resistor was the worst case error. The result was that the maximum error is 9mV when the voltage divider is not used, as the error from the input offset voltage is so dominant.

4.2.7 Power Supply Tolerance and Drift

For the first use case, the power supply tolerance will directly affect the quantized value, and the error from the power supply tolerance was considered in the quantization error. The SPX1117-L-3-3 linear regulator has a maximum output voltage of 3.366V and a minimum output voltage of 3.234V. The output voltage range is specified over the entire operation range regardless of temperature, load, or input voltage, so thermal drift and other sources of error were not considered separately, and the voltage range can be improved when lower input voltage range is used, as well as reducing the temperature of the SPX1117-L-3-3 by reducing the power dissipated. [15.]

For the second use case, a radiometric sensor was considered and is powered from the same regulator as the ADC reference voltage, so the tolerance of the power supply does not affect the conversion result. The drift due to temperature is also slow enough that it will not cause a significant error when a radiometric sensor is used.

4.2.8 Power Supply Ripple

The ripple in the power supply will also directly cause an error to a converted value. The ripple from the LT3970 switching regulator can be calculated by using the standard equation for voltage ripple of a buck DCDC based on the input voltage (V_{in}), output voltage (V_{out}), output inductance inductance (L), switching frequency (f), and output capacitor equivalent series resistance (ESR) and inductance (ESL). [19.]

$$V_{ripple(p-p)} = \frac{V_{out}(V_{in} - V_{out})}{V_{in} L f} ESR + \frac{V_{in}}{L} ESL \quad [V] \tag{18}$$

The used inductor is 22uH, the maximum input voltage is 30V, the maximum output voltage is 20V, and the switching frequency is 600kHz. A ceramic output capacitor is used, so the ESL and ESR of the capacitor is not known and may be significantly influenced by the layout. From the board and 0805 ceramic capacitor, the ESL was conservatively approximated as 4.5nH, and the ESR was conservatively approximated as 8mΩ. This corresponds to approximately 12.2mV of peak to peak ripple from the LT3970 switching regulator. [20]

The SPX1117-L-3-3 linear regulator has approximately 35dB of ripple rejection at 600kHz, which results in an output ripple of 220uV peak to peak. This is lower than the resolution of the ADC, and so will not cause a significant different on the result. However, the ripple will be considered in the calculation anyway, which means the voltage range to be considered is from 3.36622V to 3.23378V.

For the second use case the ripple of the LT3970 switching regulator was calculated with the same equation, but with a 5V output voltage instead of 20V resulting in 8.7mV of peak to peak ripple from the LT3970 switching regulator.

For this use case the SPX1117-L-3-3 linear regulator has the same characteristics resulting the an output ripple of 150uV peak to peak. The ripple will be considered as the only source of error for the linear regulator, so the voltage range of the ADC reference voltage to be considered is from 3.30015V to 3.29985V.

4.2.9 Quantization Error and Total Error

The equation for the converted value (M_{adc}) is given as a function of the signal voltage (V_{signal}), and the reference voltage (V_{ref}) for a ADC with n bit resolution ADC.

$$M_{adc} = \lfloor \frac{V_{signal}}{V_{ref}} 2^n \rceil \tag{19}$$

The error was calculated in the converted digital values, as that was used to determinate maximum accurate sensor resolution The rounding is explained by quantization error, also illustrated in Figure 15.

$$Error+ = \lfloor 2^n (\frac{V + V_{error} + V_{noise}}{V_{ref(min)}} - \frac{V_{signal}}{V_{ref(ideal)}}) \rceil \tag{20}$$

$$Error- = \lfloor 2^n(\frac{V_{signal}}{V_{ref(ideal)}} - \frac{V - V_{error} - v_{noise}}{V_{ref(max)}})\rceil \quad (21)$$

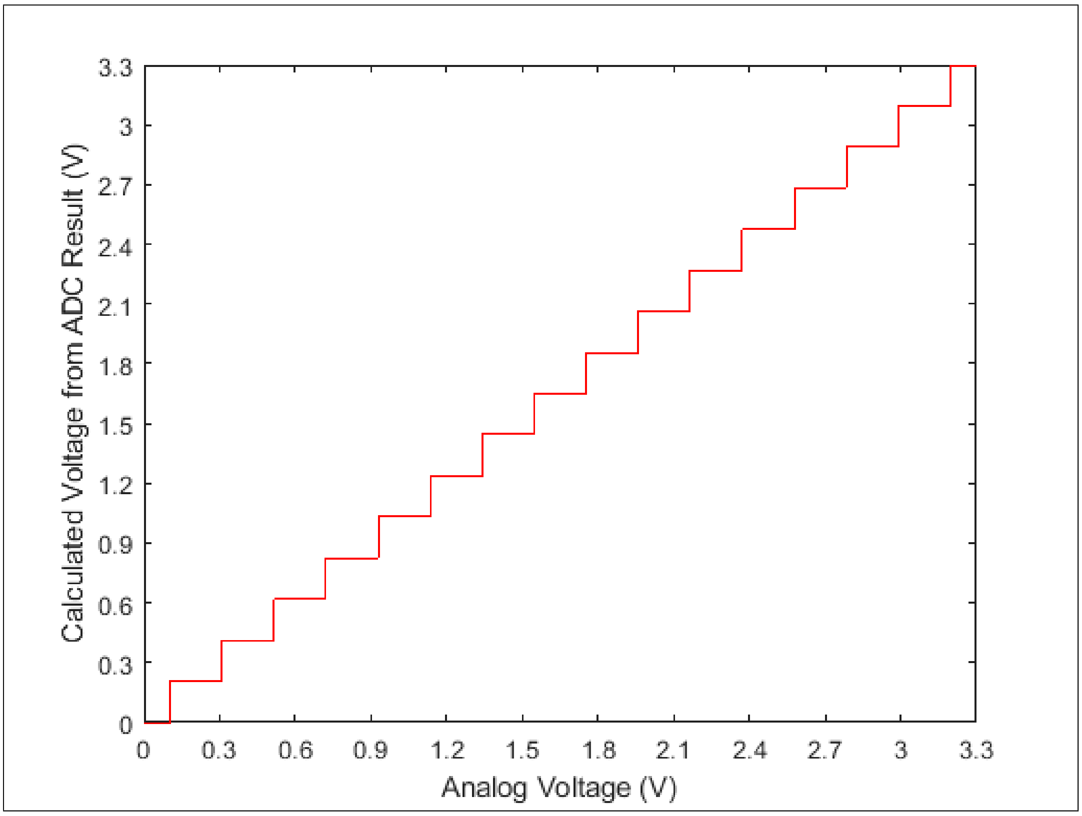


Figure 15: Illustration of quantization error by showing analog voltage vs ADC value of a 4 bit ADC.

As the error increases as the signal increases, the maximum error was calculated at 3.3V. For the first use case this gave an error of +96/-93. This error means that the accurate resolution of a single measurement is 42 steps at minimum. In reality the resolution is much better, as the input offset voltage error is a constant offset (excluding the temperature drift), but in the case that absolute accuracy is required the Analog Node must be calibrated for those specific sensors.

Doing the same calculation for the first use case if the Analog Node is calibrated in software resulted in an error of +3/-3. This error corresponds to an accurate resolution of 1365 steps, which is easily good enough for all the measurements currently made by Metropolia Motorsport.

For the second use case, as the hardware oversampling is used, the error was calculated with a 16 bit ADC. In reality the error from the ripple will usually be averaged to a much lower value, but the statistics related to that are outside the scope of this thesis. The calculated error was the worst case error, which is +3/-3. This accuracy is much better than the accuracy required for anything by Metropolia Motorsport, and corresponds to a resolution of 21845 steps. Note that the noise was not calculated here, as it must be larger than the resolution when 12 bits are used in order to achieve higher than 12 bit resolution, so calculating it without considering statistics would make any resolution higher than 12 bit completely impossible to have accurate values for.

Note that these calculations do not consider the characteristics of the sensor or measurement setup, which will usually limit the accurate resolution, especially if hardware oversampling is used.

4.3 Overcurrent Protection

On the Power Nodes there were two important current overprotection features. The first was the overcurrent protection for everything powered by that Power Node, as that is the main purpose of the Power Nodes. The second was the fusing for the Power Node itself.

4.3.1 Wiring and Supplied Device Overcurrent Protection

The main purpose of the Power Nodes is overcurrent protection. A fuse is also used to protect the LV power supply, so the Power Node should react faster than the fuse. This corresponds to protecting the wiring, as the fuse is sized for the wiring. Unfortunately, the specific fuse used is not known, but it is some automotive blade mount fuse (for example, see Figure 16), and a 10A fuse was used in the last Metropolia Motorsports car, with wiring rated for 10A.

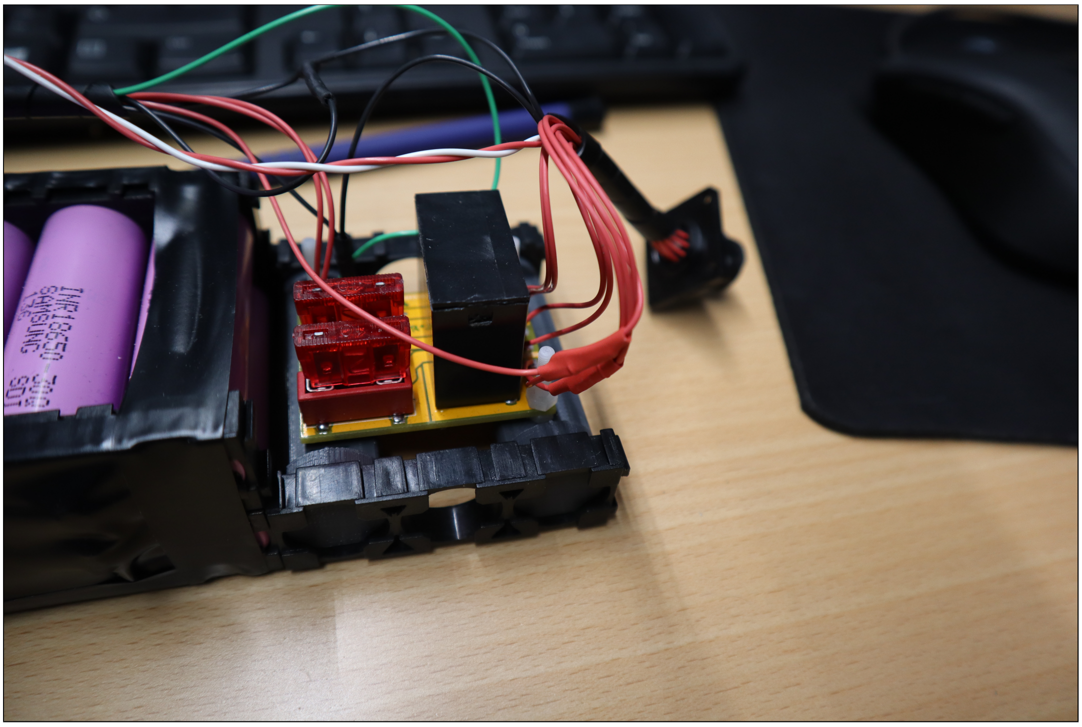


Figure 16: main Fuse for LV of HPF019, Metropolia Motorsports car from 2019.

There is higher current available when the LV supply is a battery than when the LV supply is a DCDC, so the maximum current output was checked considering a battery. In this case, the internal resistance of the battery is the dominant factor limiting the short circuit current. Lithium-ion was considered, as this is currently the chemistry with the best power density, which was approximated as having 20mΩ per cell based on INR18650-30Q, the cells used for the last LV battery [21]. Each cell has a maximum of 4.2V (though the voltage drops incredibly quickly under load [21]. This voltage and internal resistance gives a maximum short circuit current of 210A. As past batteries (see Figure 17 for example) used by Metropolia Motorsport have used three cells in parallel 630A was considered, though only for later selection of the fuse, as the used high side switch clamps the current.

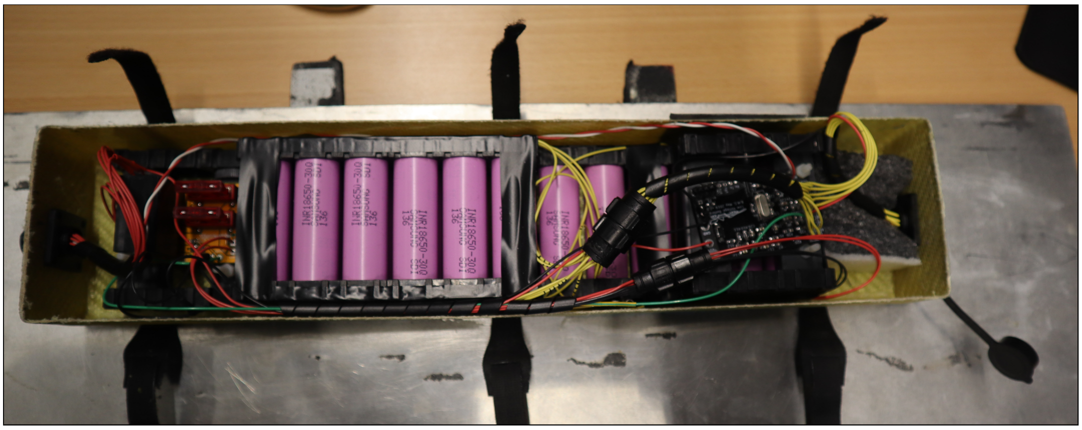


Figure 17: LV battery of HPF019, Metropolia Motorsports car from 2019.

Some approximation of fuse I^2t is required to calculate how fast the Power Node must switch off. The fastest 10A blade mount automotive fuses that Littelfuse sells are currently in the MICRO2 series, which has an I^2t of 89, so this value was used as the approximation of the I^2t rating of the LV main fuse [22]. The used high side switch clamps the current at 90A at most, so this was used as the upper limit of overcurrent that needs to be protected by software [6]. This means the Power Node must be faster than 11.0ms, as the time for Power Node to open a switch is not dependant on the current.

There are several delays from the VND7020AJ high side driver that will reduce the maximum amount of time for the microcontroller to switch it off. The first is the transition time for the multiplexed outputs of the VND7020AJ high side driver of 20us maximum. This will need to be taken into account when considering the microcontroller ADC conversion time. The other delay from the VND7020AJ high side driver is the turn off time, which is at most 100us. [6.]

The STM32G441 microcontroller has two ADCs, but one of them is used to measure the voltage across the reverse protection diode to subtract from the measured value from each VND7020AJ. As there are at most 3 VND7020AJ high side drivers per microcontroller, each with two channels, one ADC on the STM32G441 microcontroller must convert 6 values to measure the current of everything. Additionally, the voltage and temperature of each VND7020AJ high side driver should be measured. This results in a total of 12 values to read per cycle and necessitates 4 multiplexing changes, each requiring a 20us wait.

Fortunately, the ADC conversions is run with DMA, so whatever else the microcontroller is doing during the conversion process was not considered. Each current is read once every time at most twelve conversions are performed and 4 waits are executed. Unfortunately the 20us specified on the VND7020AJ datasheet was too short, so 100us wait time is used instead. Similarly, while a sampling time of 247.5 cycles (corresponding to 791.67ns) should be enough according to the microcontroller datasheet, the measurement becomes much more stable with a sampling time of 640.5 cycles at a lower clock speed (60MHz) (corresponding to 10.675us) [2]. As the ADC takes 167ns for a 10 bit conversion with a 60MHz clock, the total time between measurements is 530.104us.

Fortunately, this means that overcurrent protection is still easy, even though the current monitoring is much slower than expected, a 20 sample rolling average (10.062ms) is still faster than the fuse.

4.3.2 Power Node Fusing

As the Power Node is the device in the car protecting other devices from overcurrent, the Power Node does not have overcurrent protection. Therefore, a fuse was placed on the Power Node so that in the event of damaged components on the Power Node, the main LV fuse does not blow.

The fuse was also sized to protect the traces on the PCB, so that the damaged Power Node can be repaired by swapping components. The maximum I^2t for this can be calculated from the k^2S^2 of the fuse where S is the cross sectional area of the conductor and k is some constant based on the volumetric heat capacity of the conductor at 20°C (Q_c), the reciprocal of temperature coefficient of resistivity at 0°C (β_0), the resistivity at 20°C (ρ_{20}), the initial conductor temperature (θ_i), and the maximum temperature of the conductor (θ_f) [23].

$$I^2t \leq k^2S^2 \tag{22}$$

$$k = \sqrt{\frac{Q_c(\beta_0 + 20)}{\rho_{20}} \ln \frac{\beta_0 + \theta_f}{\beta_0 + \theta_i}} \tag{23}$$

The copper will have 1oz thickness, which corresponds to .0347mm of thickness, and the smallest traces to protect are .254mm wide, meaning the cross sectional area is .0088138 mm^2. Volumetric heat capacity of copper is approximately .003449600J/mm^3°K, reciprocal of temperature coefficient of resistivity of copper is approximately 254°K, and resistivity of copper is approximately $1.68*10^{-5}$Ωmm. The initial temperature is approximately the same as the maximum ambient temperature, being 323.15°K. The maximum temperature is dependant on the PCB material, but a conservative estimate for FR-4 if the manufacturer is not known is 393.15°K (120°C).

Using Equation 23, it was calculated that the maximum acceptable I^2t of the fuse is .5A^2/s.

5 Software

The code for the Power Nodes [1] and Analog Nodes [2] will only be described at a high level and very briefly, but as they are on GitHub the code can be obtained if the reader wishes to view the code in more detail. The code that is primarily of interest is in the Src and Inc folders except system_stm32g4xx.c, and stm32g4xx_hal_conf.h.

5.1 Power Node Code

A high level block diagram of the Power Node code is shown in Figure 18 and Figure 19. The focus of the code was to check the current and switch off any channels with too high current as soon as possible, as well as to put a priority for sync CANbus messages. CANbus messages are only sent from the main to avoid rx header corruption. In the code for both Power Nodes and Analog Nodes, errors are set to warn of possibly unwanted behavior or of invalid configuration of a device. For example, if a Power Node is configured to send a CANbus message with a Data Length Code (DLC) of 0 or an invalid identifier it will send an error.

5.2 Analog Node Code

A high level block diagram of the Analog Node code is shown in Figure 20 and Figure 21. This is simpler than the Power Node, as the Analog Node only does analog conversions and sends values on CANbus, while the Power Node also has digital and PWM IO, as well as control of high side drivers. As there are fewer CANbus messages to send, sync messages are not considered any different than regularly sent CANbus messages sent at a set interval. However, CANbus configuration has not been written for the Analog Node due to time constraints.

[1] Power Node GitHub repository is found here: https://github.com/MetropoliaMotorsport/powerNode
[2] Analog Node GitHub repository is found here: https://github.com/MetropoliaMotorsport/AnalogNode

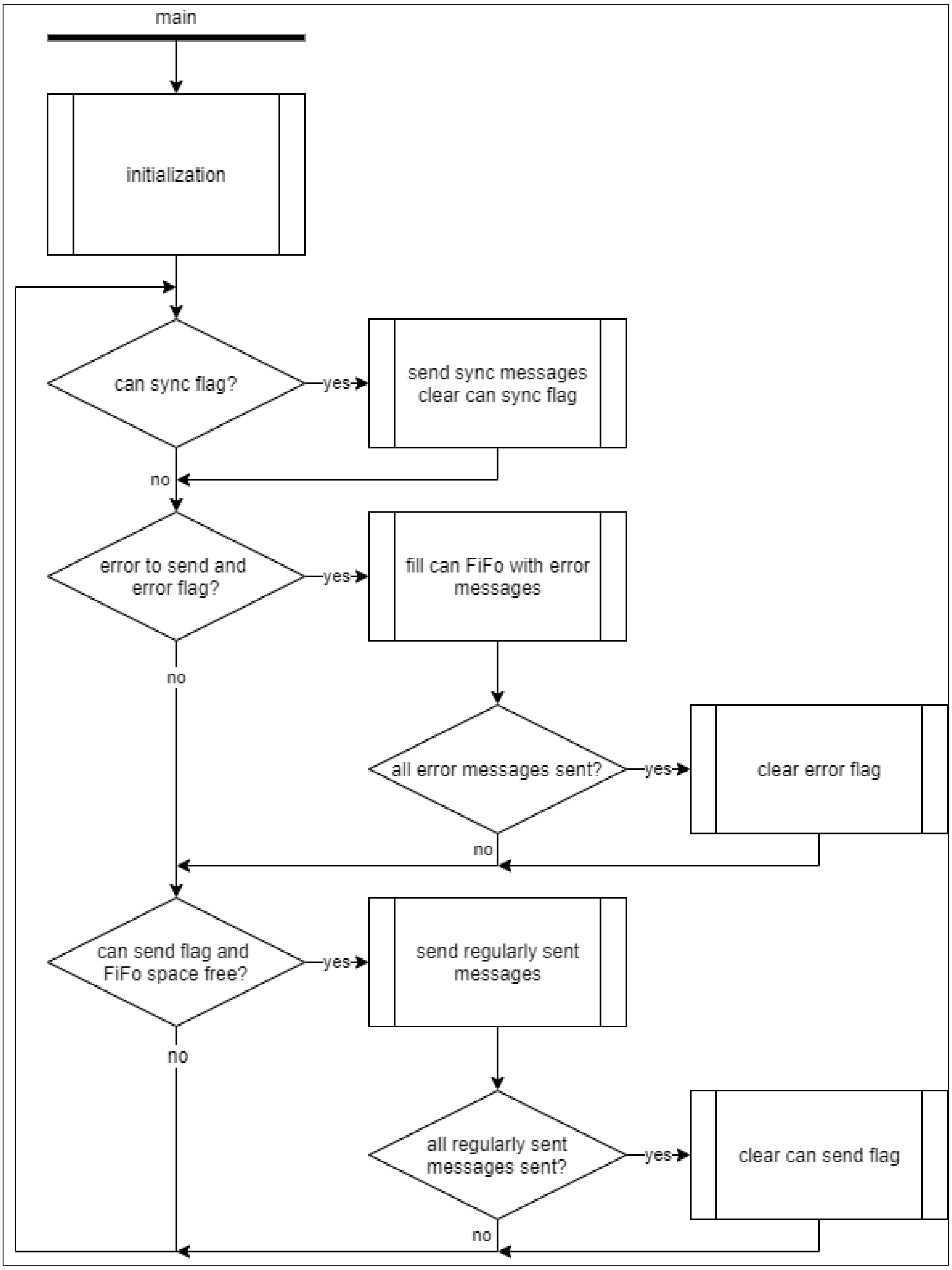


Figure 18: Power Node high level block diagram of main loop

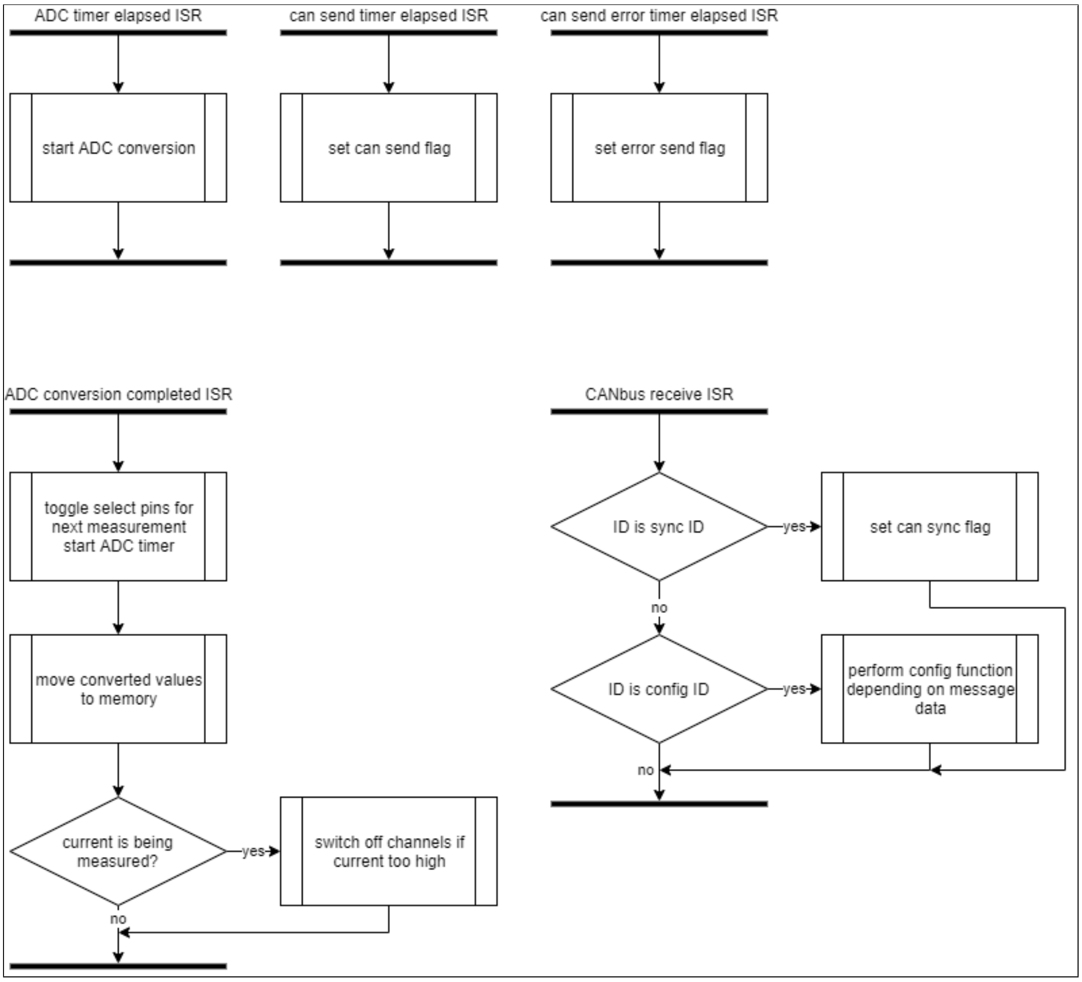


Figure 19: Power Node high level block diagram of interrupts

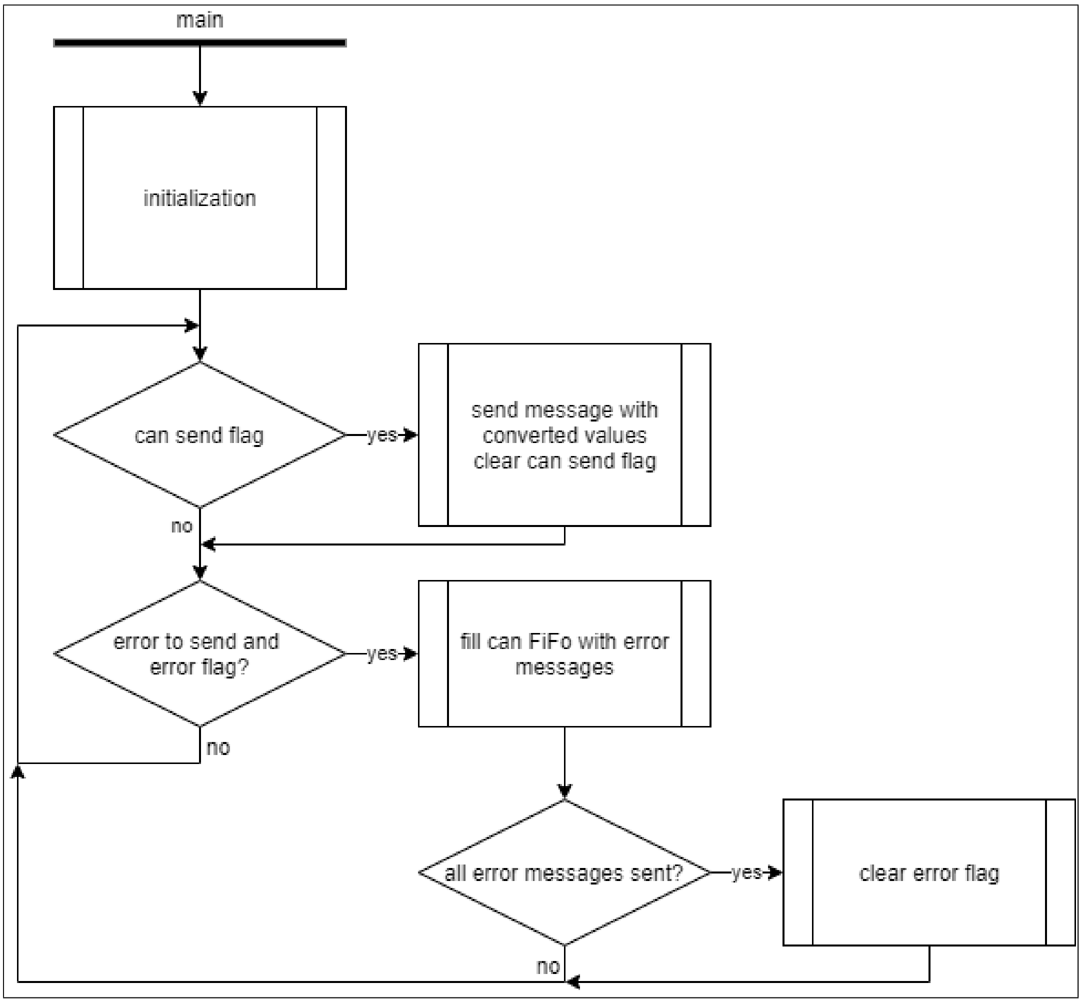


Figure 20: Analog Node high level block diagram of main loop

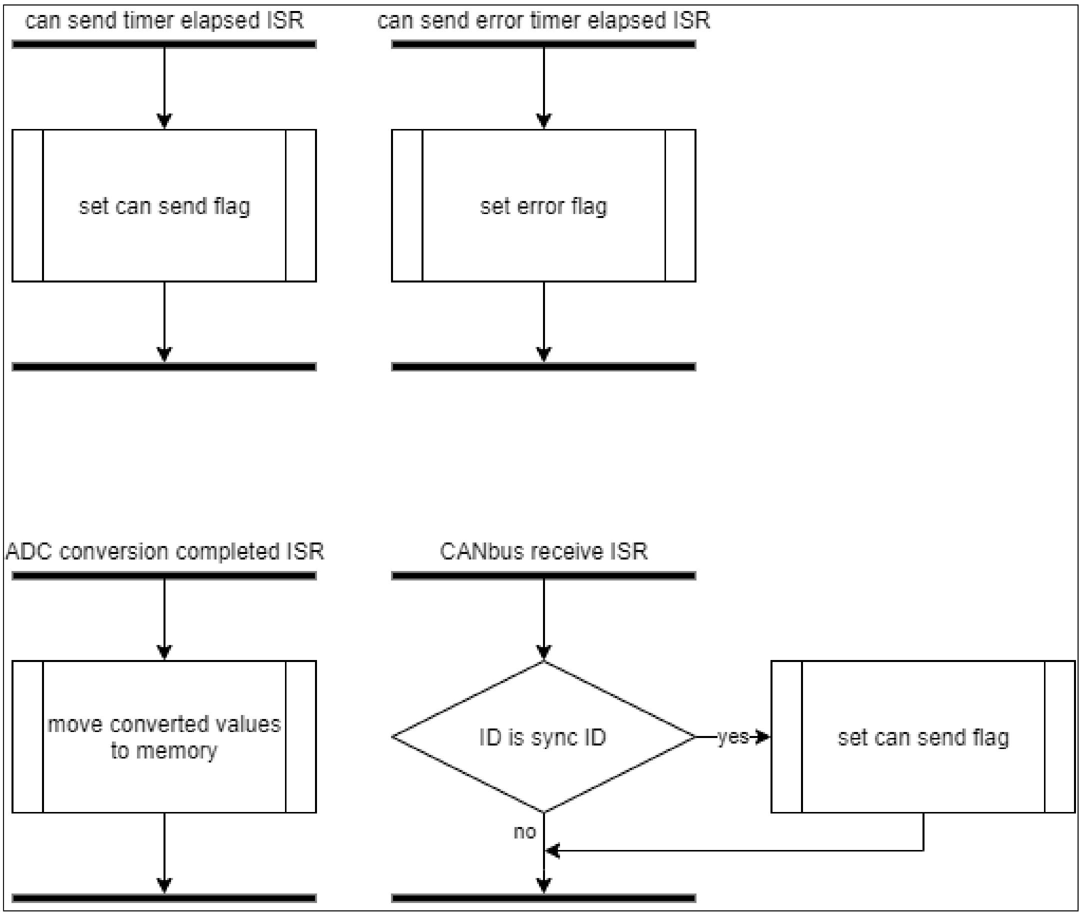


Figure 21: Analog Node high level block diagram of interrupts

6 User Manual

Manuals were written for assembling, testing, and using both nodes. These manuals are intended for Metropolia Motorsport team members who already have an understanding of basic electronics. The assembly and testing parts require also the ability to solder, understanding of how to use a multimeter, and knowledge of how to put code on an STM32 with an ST-link, as well as ability to use Kvaser CanKing to view CANbus messages on a computer. The CANbus configuration part is intended for someone already capable of programming and having a good understanding of CANbus. The pinout part is intended for someone knowledgeable about programming the STM32 microcontroller with their tool of choice, for example, STM32CubeIDE and the HAL libraries.

The manuals are included in Appendix 1 and Appendix 2

The first part of the manual is assembly and testing instructions. The focus of this was to instruct someone who has not assembled either type of board before as to how to assemble it and determine faults so that it is easier to debug faults. However, the target was not to train someone to actually debug the faults, but only to be able to find them before any PCBs are used. Also described was how to assemble some components for different uses of IO pins (different voltage dividers for Analog Node or PWM IO instead of digital input for Power Node).

The assembly section also has a recommended connector pinout section aimed towards whoever is designing the wiring harness. Whoever is assembling the nodes must confirm that this was followed in the wiring harness and note for the programmer which IOs are used for what purposes in the car.

The pinout section was targeted towards future programmers who either need to modify the existing code, or who need to rewrite the code entirely for some reason.

The CANbus configuration section (currently only for the Power Node) has two purposes, the first of which was to be a reference guide for programmers of other devices in the car in order to determine what CANbus messages they will send to control the nodes. The second purpose was to act as a reference for someone configuring a node over CANbus (or someone writing software to do this automatically).

7 Measurements and Validation

There were many things that should be checked and validated on the completed boards. However, there was only time to check some of them so the most important things were tested. These consisted of checking the accurate resolution on the Analog Node ADC, checking that the high side drivers on the Power Nodes can handle reasonable inductive loads and enough current continuously, and checking the current sense on the Power Nodes. Basic things like checking that the microcontroller are capable of sending CANbus messages or that DCDC output voltages are correct are not discussed here, but they are also checked.

7.1 Analog Node ADC

The 12 bit accuracy of the ADC was compared to HP 3468A digital multimeter, which has resolution (and accuracy) much higher than we are aiming for with the 12 bit input of the Analog Node. A potentiometer was used for the input (powered from the 3.3V supplied by the Analog Node, but the accuracy of the linear regulator still factored in as absolute voltage measurements were taken instead of potentiometer position measurements), with the setup shown in Figure 22. [24.]

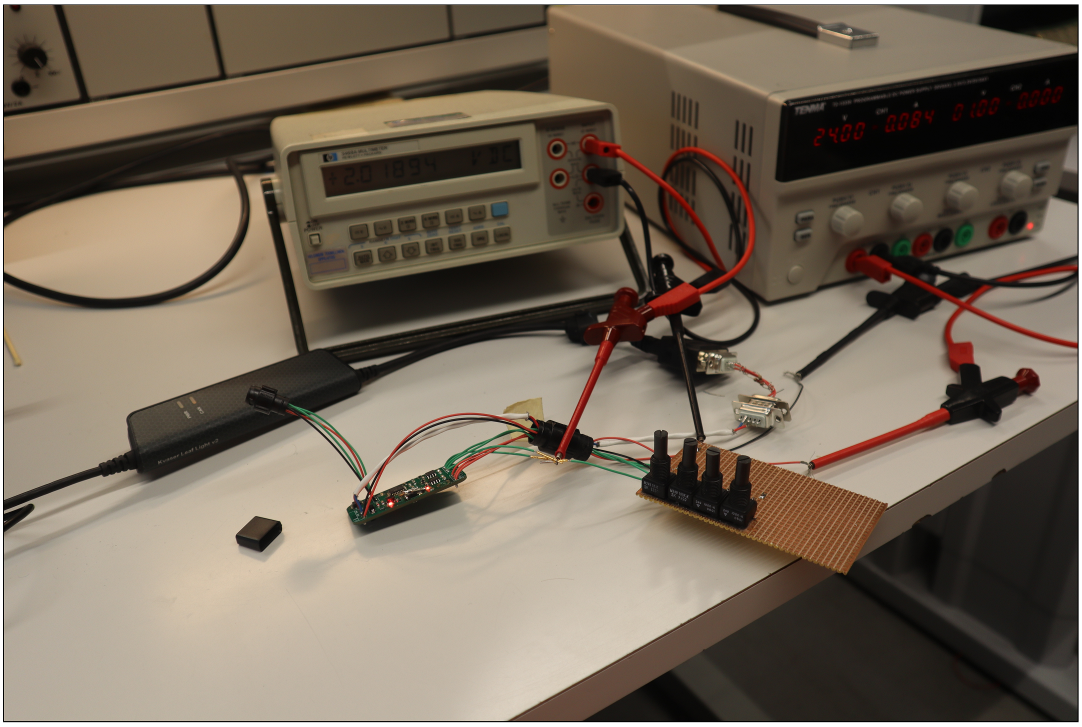


Figure 22: Setup for testing Analog Node input accuracy.

Values were read from the multimeter and compared with the values calculated by the Analog Node (no calibration is done). One input with no voltage divider (capable of reading up to 3.3V) was checked, as was an input with a voltage divider intending for 24V input voltage, and the results are shown in Table 2 to the nearest mV.

Table 2: Analog Node analog input comparison with multimeter.

HP 3468A (V)	3.3V Input (V)	Error (mV)	24V Input (V)	Error (mV)
.001	0	1	0	1
.138	.132	6	.096	42
.202	.197	5	.163	39
.312	.307	5	.271	41
.406	.401	5	.368	38
.520	.515	5	.476	44
.611	.607	4	.573	38
.708	.703	5	.669	39
.806	.801	5	.772	34
.918	.913	5	.880	38
1.041	1.037	4	1.007	34

1.110	1.105	5	1.073	37
1.210	1.206	4	1.175	35
1.317	1.313	4	1.284	33
1.409	1.405	4	1.374	35
1.503	1.500	3	1.471	32
1.608	1.605	3	1.573	35
1.703	1.701	2	1.670	33
1.818	1.817	1	1.790	28
1.935	1.934	1	1.905	30
2.019	2.020	1	1.989	20
2.109	2.111	2	2.080	29
2.200	2.203	3	2.170	30
2.314	2.317	3	2.284	30
2.401	2.403	2	2.375	26
2.517	2.520	3	2.489	28
2.622	2.625	3	2.592	70
2.705	2.709	4	2.676	29
2.815	2.819	4	2.785	30
2.914	2.919	5	2.887	27
3.024	3.029	5	3.996	28
3.115	3.114	1	3.086	29
3.200	3.167	33	3.177	23
3.288	3.196	92	3.261	27

For the input without a voltage divider this looks very good except for the last two measurements as the error is less than just the input offset voltage of the LM358 operational amplifier [11]. For the measurements done with a 24V voltage the result was well within the calculated error, and if calibrated would be within the determined error considering that the resolution is 6mV because of the voltage divider.

The error from the last two measurements was a voltage drop across the damping resistor (R2 in Figure 14). I suspect this was due to leakage current through the internal protection diodes of the STM32G441, as shorting the resistor removed the error from the measurement. Unfortunately, there are no details about the protection diodes in the microcontroller datasheet, so in the case that the entire ADC range must be used, the damping resistor should be shorted (or in the case of a voltage divider, smaller resistances should be used), and the inductor or capacitor values should be reduced (or one filtering component should be removed). [2]

Unfortunately at the time of writing this thesis there has not been time to test the hardware oversampling method of improving the resolution, so 16 bit analog inputs are not discussed here. However, given that the individual sampled values were changing, it seems very likely that the hardware oversampling will work. This is likely caused by external noises, interferences (including some internal to the microcontroller), and ripples not accounted for in the calculations, possibly even some error coming from the failing mechanical connection of the used potentiometer.

7.2 Power Node Current Sense

The testing of the Power Node current sense was done at the same time as calibration. No uncalibrated calculation was done, as it is very unclear from the datasheet what ranges different values would be valid for, so it was easier to feed current through the VND7020AJ high side driver and check the resulting raw values from the ADC [6]. Two channels (from different high side drivers) were connected in series with a power trimmer (see Figure 23), and a Mastech MY64 multimeter to measure the current [25]. The raw value of the analog input used for current sensing and the value read by the multimeter was recorded in Table 3 and Table 4. The raw values were taken as a time average, rounded to the nearest .5.

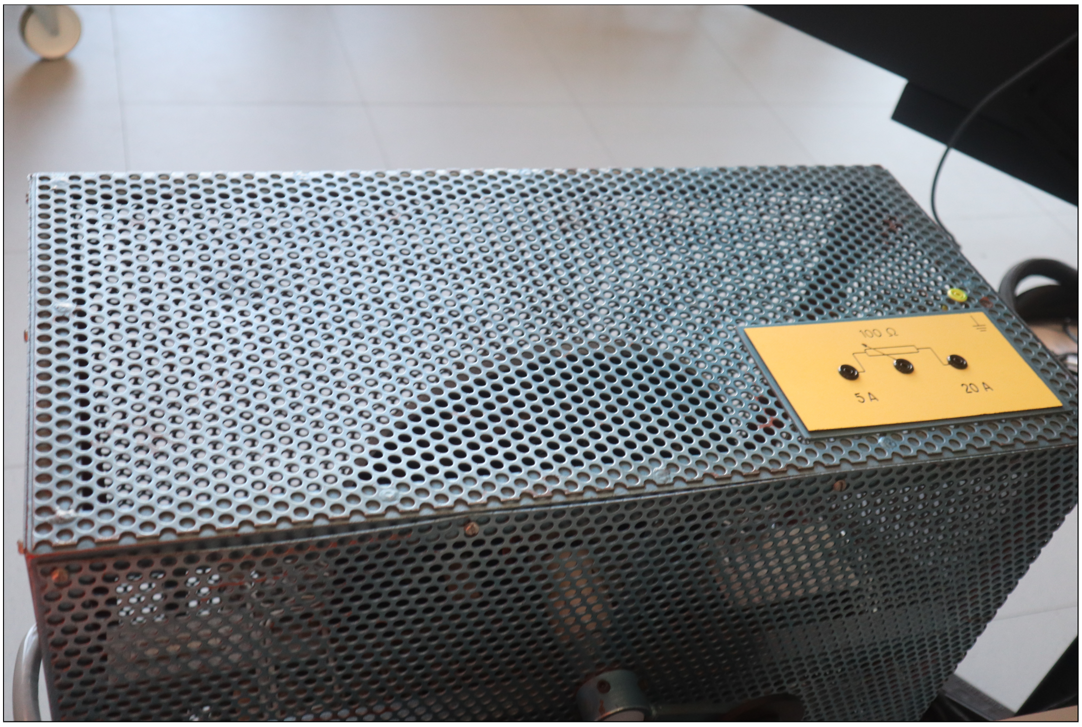


Figure 23: Power trimmer used for testing high side drivers.

Table 3: 1st channel current sense comparison with multimeter.

Mastech MY64 (A)	Raw Value	Calculated Value (A)	Error (A)
0	178	.05	.05
.27	188.5	.36	.09
.4	194	.52	.12
.7	205.5	.85	.15
1.03	218.5	1.23	.2
4.21	327.5	4.38	.17
4.92	355	5.18	.26
5.25	362	5.38	.13
5.72	376	5.79	.07
6.15	392	6.25	.1
6.47	403	6.57	.1
6.65	406.5	6.67	.02
7.11	422.5	7.13	.02
7.56	435	7.49	.07

Table 4: 2nd channel current sense comparison with multimeter.

Mastech MY64 (A)	Raw Value	Calculated Value (A)	Error (A)
0	166	-.29	.29
.27	176.5	.01	.26
.41	183	.20	.21
.75	198	.63	.12
.99	209	.95	.04
1.38	224	1.38	0
4.21	332	4.51	.3
4.62	344	4.86	.24
5.09	358	5.26	.11
5.35	364	5.44	.09
5.91	380	5.90	.01
6.21	386	6.07	.14
6.66	398	6.42	.24
7.04	408	6.71	.33
7.53	419	7.03	.5

These measured values were used to make a line of best fit, being $I = .02895 * raw - 5.1$ with I being current in amps. The results of this from the previous raw values are included in Table 3 and Table 4 in the third column as **Calculated Value (A)**.

Unfortunately, the results were not very accurate, having an error of +.5/-.29A. However, this is easily accurate enough for the purpose of checking if a device is broken and removing power from that device before a fuse blows. Some discussion of the reasons for this large error and possible improvements are discussed in 8.1.3.

7.3 Power Node High Side Driver

The final important thing to test was done at the same time as testing the current sense of the Power Node. The Power Node was put in heatshrink (as seen in Figure 24) and let continuously run current for several minutes. Fortunately, this test went very well, and even with the maximum available current at the time of 8.3A, the Power Node did not go into thermal shutdown, and the maximum external surface temperature of the heatshrink was approximately 75°C measured with a Fluke TiS75 infrared camera [26]. Unfortunately the internal chip temperature sense was not working at the time, so the internal temperature is not known, but this is far more current than was expected.

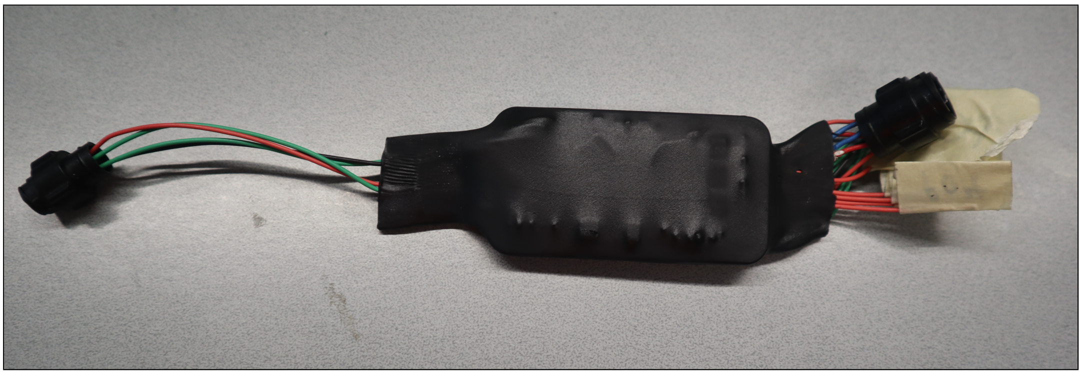


Figure 24: Power Node in heat shrink, as it will be in the car for environmental protection.

While the current was only over in one channel, all other channels were active and the power dissipation due to carrying current can be approximated as resistive, meaning if the 8.3A is used as a limit for the maximum power driven by all six channels, the square sum of currents may be at least 68.9A^2.

8 Conclusions and Future Developments

This thesis project was about the design of two types of PCBs for a Formula Student car. One measures analog signals and sends them to CANbus (called the Analog Node) and the other acts as a programmable fuse and CANbus controlled switch board (called the Power Node).

8.1 Future Improvements

While the project successfully achieved all of its goals, there are some places for improvement depending on what is needed in future Metropolia Motorsports cars.

The main improvements would be to allow configuration of the Analog Nodes over CANbus, consideration of a CANbus bootloader and/or Real Time Operating System (RTOS), implementation of hardware oversampling for the Analog Nodes (and possibly improved analog sensor resolution if hardware oversampling is insufficient), and better accuracy of current sense on the Power Nodes.

8.1.1 Software Improvements

After the completion of this thesis if there is time I will try to implement these features, so check the newest version of the code (github linked in 5) if you are developing this project further.

The largest missing feature was that the Analog Node currently does not read or write any settings to flash, and so can not be configured over CANbus right now. A smaller improvement for both nodes would be to detect undervoltage, as the hardware exists for undervoltage detection, but the software does not.

Another possible improvement would be to implement CANbus bootloaders in all the microcontrollers in the car. This would make saving configuration in the flash redundant and would make updating code easier. Similar to this, there has been some discussion of implementing a RTOS, which could also improve some features of the software, but would require a complete rewrite of the software.

8.1.2 Analog Node Higher Resolution and Accuracy

In the case that higher resolution than 12 bit is required for an analog sensor in a future Metropolia Motorsports car, the hardware oversampling should be tested early in the season. This is also primarily a software change, but in the event that the system is not accurate enough, new hardware will need to be designed.

I would recommend using an external ADC in that case, as they are very easy to get even relatively cheap and small ADCs that have much better resolution, accuracy, and conversion speed than the ADCs integrated into microcontrollers. An external ADC also is not as affected by the interference generated by the microcontrollers, which is a possible major source of error, and a possibly gives the option of using separate analog supply voltage and analog reference voltages, so that the voltage reference can be incredibly stable. It could also mean that voltage dividers do not need to be used, but this depends on the specific ADC chosen. In this case all the errors considered in this thesis should be considered, as well as minimizing the input resistance to the ADC, which will require modifications to the lowpass filtering described in 4.1.

Another change to improve the accuracy would be to do the DCDC layouts as one sided layouts, though this takes more space, an improved layout would reduce the ripple of the DCDC significantly. If there is interest in this, I recommend reading Understanding and Minimising ADC Conversion Errors from STM and especially the LT137 datasheet [17] [19]. I would also recommend watching a video of the 2018-2019 LTSpice Seminar by Mike Engelhardt, as it contains good information about buck regulators[3].

Another consideration with the ripple would be filtering the harmonics, as the ripple rejection for linear regulators is poor at high frequencies, and even the SPX1117-L-3-3 linear regulator does not specify ripple rejection in the MHz range [15]. I would also suggest

[3] Video of LTSpice Seminar is found here: https://yashrk.wordpress.com/2018/11/10/ltspice-seminar-by-mike-engelhardt/

looking up other information about filtering ripple, for example, capacitance multipliers [4]. One consideration would be to not use a switching regulator at all, but voltage input filtering should still be considered, especially if the LV system is not supplied from a battery.

8.1.3 Power Node Current Sense Improvements

If a Power Node with better current sense is desired for some reason, a new design will need to be done. The first part of this would be to choose a high side driver with a more accurate current sense. One large problem with the used high side driver is that under a fault it will output 6.6V, so that must not break the microcontroller, but it also has a huge current range (well over the maximum current it should ever see in a Metropolia Motorsport car), meaning there is a very low resolution. The resolution could also be improved by amplification or a lower reference voltage, but as most high side drivers with current sense have such a high current measurement range and maximum current sense output current / voltage, it would almost certainly require the measurement system to be able to handle a continuous overvoltage condition.

The VND7020AJ high side driver was chosen because it also had an internal temperature sensor and voltage sense that can be multiplexed between, but this introduces other problems [6]. Multiplexing takes time, but more importantly, external multiplexed resistors should be used in order to be able to measure current, voltage, and temperature, as each would require a different resistance and overvoltage protection for the microcontroller.

If possible a high side driver should be chosen that does not require the ground protection network required by the VND7020AJ high side driver, as the measurement for the voltage offset also introduces some software complexity [6]. In the Power Node the measurement is not used as it is not consistent even under conditions when it should be, likely due using the analog input pin adjacent to the oscillator input pin. I would recommend not using the pins adjacent to the oscillator input pin as analog inputs, as that could be a significant source of interference.

All of these changes will require a bit larger microcontroller to be used as well, so the software will have to be either entirely rewritten or at least modified enough to run on the

[4] Video about Capacitance Multiplier is found here: https://www.youtube.com/watch?v=wopmEyZKnYo

newly chosen microcontroller as well.

8.2 Conclusion

So far the project has been very successful, considering it is a prototype. In the near future, the boards will be used and problems with them may be discovered, but with all testing so far the boards will work much better than necessary where it is important.

In the near future, both boards with be used in Metropolia Motorsports newest car, which may result is new software features being desired or software bugs being discovered that were not considered during the testing phase. There may also be problems needing to be fixed related to interference, for example from the inverters or on the LV power supply, or other hardware related problems. However, with the testing so far I expect there will be no major issues using the Power Nodes and Analog Nodes in the coming months.

The Power Node is capable of being configured by CANbus, can switch off faster than a blade mount fuse, and can carry much more current than was originally expected. The Analog Node is capable of sufficient measurements and calculations of all sensors currently used in Metropolia Motorsports cars. The assembled Analog Nodes for Metropolia Motorsports 2020 car are shown in Figure 25 and Figure 26, and the Power Nodes for the same car are shown in Figure 27 and Figure 28.

Figure 25: Top of assembled Analog Nodes.

Figure 26: Bottom of assembled Analog Nodes.

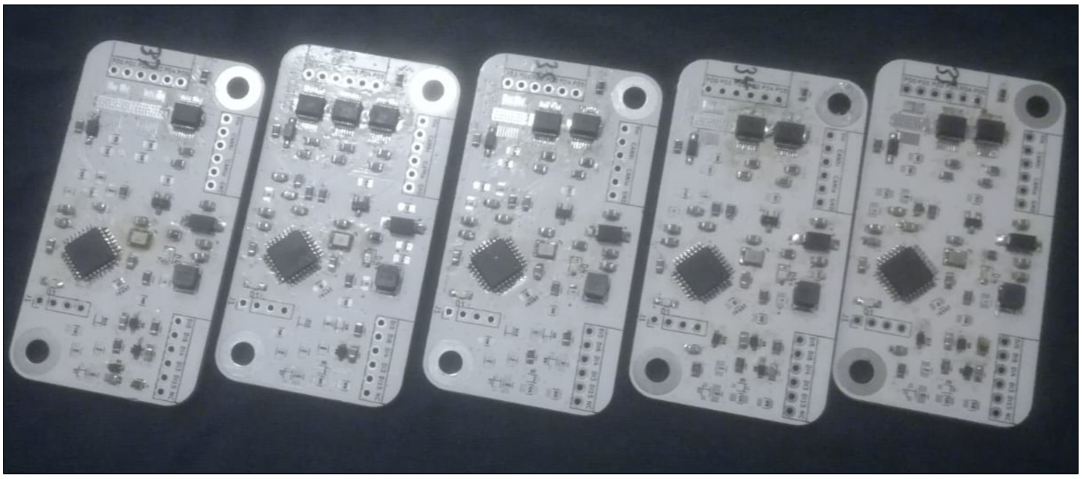


Figure 27: Top of assembled Power Nodes.

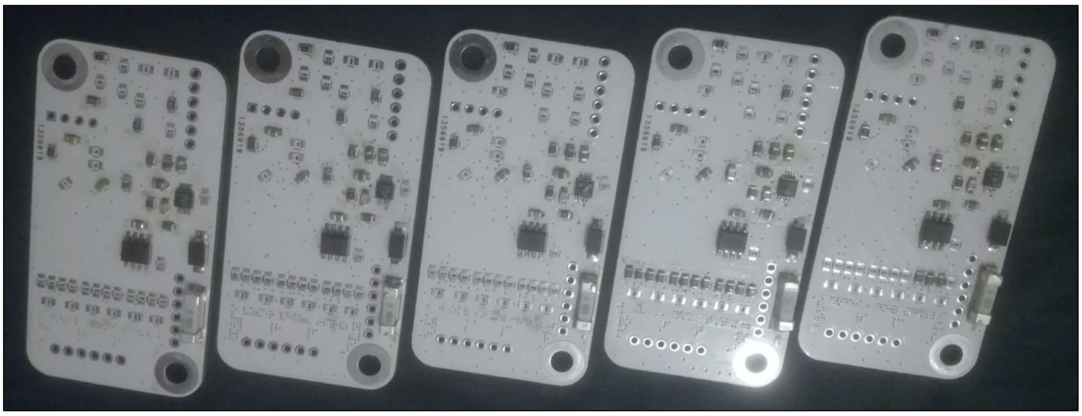


Figure 28: Bottom of assembled Power Nodes.

Bibliography

1 ISOMETER® IR155-3203/IR155-3204. Bender; 2018. Available from: https://www.bender.de/fileadmin/content/Products/d/e/IR155-32xx-V004_D00115_D_XXEN.pdf [cited February 5, 2020].

2 STM32G441xB. STMicroelectronics; 2019. Available from: https://www.st.com/resource/en/datasheet/stm32g441vb.pdf [cited October 24, 2019].

3 Power PROFET. Infineon Technologies AG; 2019. Available from: https://www.infineon.com/cms/en/product/power/smart-low-side-high-side-switches/automotive-smart-high-side-switch-profet/power-profet/ [cited October 24, 2019].

4 Smart power switches – Products. Texas Instruments; 2019. Available from: http://www.ti.com/power-management/power-switches/smart-power-switches/products.html [cited October 24, 2019].

5 High Side Switches. STMicroelectronics; 2019. Available from: https://www.st.com/en/automotive-analog-and-power/high-side-switches.html [cited October 24, 2019].

6 VND7020AJ. STMicroelectronics; 2017. Available from: https://www.st.com/resource/en/datasheet/vnd7020aj.pdf [cited October 24, 2019].

7 VN7140AJ, VN7140AS. STMicroelectronics; 2019. Available from: https://www.st.com/resource/en/datasheet/vn7140aj.pdf [cited October 24, 2019].

8 SN65HVD23x 3.3-V CAN Bus Transceivers. Texas Instruments; 2018. Available from: http://www.ti.com/lit/ds/symlink/sn65hvd230.pdf [cited October 24, 2019].

9 Robb S. CAN Bit Timing Requirements. NXP Semiconductor; 1999. Available from: https://www.nxp.com/docs/en/application-note/AN1798.pdf [cited October 22, 2019].

10 ECS-3225MV. ECS Inc International; 2019. Available from: https://ecsxtal.com/store/pdf/ECS-3225MV.pdf [cited October 24, 2019].

11 LM158, LM258, LM358, LM158A, LM258A, LM358A. STMicroelectronics; 2017. Available from: https://www.st.com/resource/en/datasheet/lm358.pdf [cited October 25, 2019].

12 ADA4522-1/ADA4522-2/ADA4522-4. Analog Devices; 2017. Available from: https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4522-1_4522-2_4522-4.pdf [cited October 25, 2019].

13 Standard Resistor Values. EETech Media; 2019. Available from: http://www.resistorguide.com/resistor-values/ [cited October 24, 2019].

14 LT3970 Series. Analog Devices; 2009. Available from: https://www.analog.com/media/en/technical-documentation/data-sheets/3970fc.pdf [cited October 25, 2019].

15 SPX1117. MaxLinear; 2018. Available from: https://www.maxlinear.com/ds/spx1117.pdf [cited October 25, 2019].

16 Type 0678H Enhanced-breaking Capability Brick Fuse. Bel Fuse; 2019. Available from: https://www.belfuse.com/resources/datasheets/circuitprotection/ds-cp-0678h-series.pdf [cited November 7, 2019].

17 UNDERSTANDING AND MINIMISING ADC CONVERSION ERRORS. STMicroelectronics; 2003. Available from: https://www.st.com/content/ccc/resource/technical/document/application_note/9d/56/66/74/4e/97/48/93/CD00004444.pdf/files/CD00004444.pdf/jcr:content/translations/en.CD00004444.pdf [cited October 27, 2019].

18 MT-047 TUTORIAL. Analog Devices; 2008. Available from: https://www.analog.com/media/en/training-seminars/tutorials/MT-047.pdf [cited October 30, 2019].

19 LT1375/LT1376. Analog Devices; 1995. Available from: https://www.analog.com/media/en/technical-documentation/data-sheets/13756fd.pdf [cited February 20, 2020].

20 Roay T, Smith L, Prymak J. ESR and ESL of Ceramic Capacitor Applied to Decoupling Applications. Kemet Electronic Corporation. Available from: http://www.kemet.com/Lists/TechnicalArticles/Attachments/91/Sun%20Paper%20on%20ESL%20and%20ESR.pdf [cited November 6, 2019].

21 Introduction of INR18650-30Q. Samsung SDI. Available from: https://eu.nkon.nl/sk/k/30q.pdf [cited February 20, 2020].

22 MICRO2™ Blade Fuses Rated 32V. Littelfuse. Available from: https://www.littelfuse.com/~/media/automotive/datasheets/fuses/passenger-car-and-commercial-vehicle/blade-fuses/littelfuse_micro2_datasheet.pdf [cited November 7, 2019].

23 Calculation of k Factor. myCableEngineering. Available from: https://mycableengineering.com/knowledge-base/calculation-of-k-factor [cited November 7, 2019].

24 Digital Multimeter 3468A. Agilent; 1982. Available from: https://literature.cdn.keysight.com/litweb/pdf/03468-90004.pdf?id=805908 [cited February 9, 2020].

25 MY64 - Digital Multimeter. Mastech. Available from: http://www.mastech-group.com/products.php?PNo=155 [cited February 10, 2020].

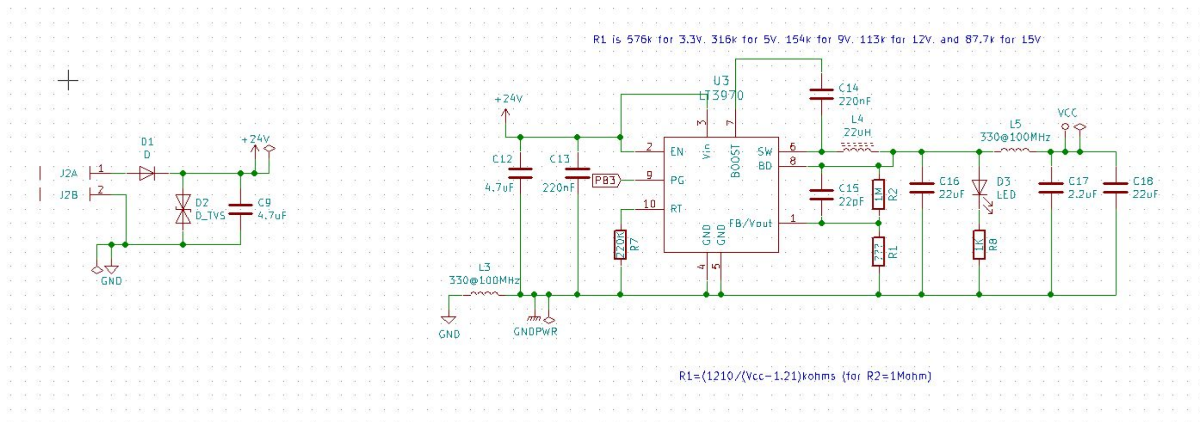
26 Fluke TiS75 Infrared Camera. Fluke. Available from: https://www.fluke.com/en-us/product/thermal-cameras/tis75# [cited February 10, 2020].

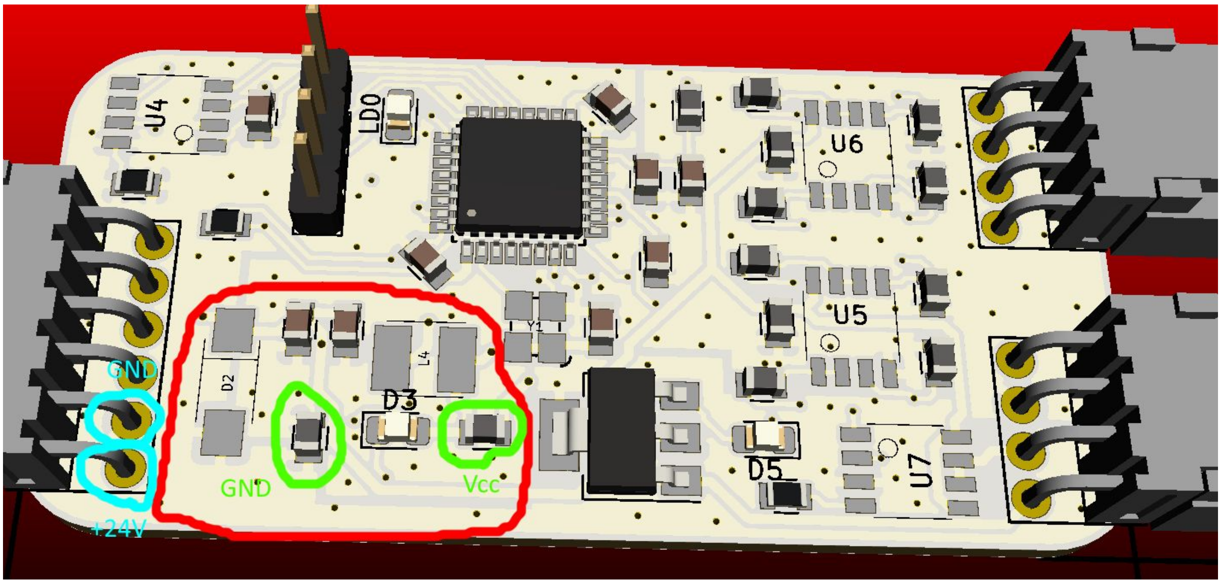
Analog Node 2020 Assembly and Programming Guide

Assembly and Hardware Testing Instructions

First, check that all four layers are present on the board with a light on the place shown in the images.

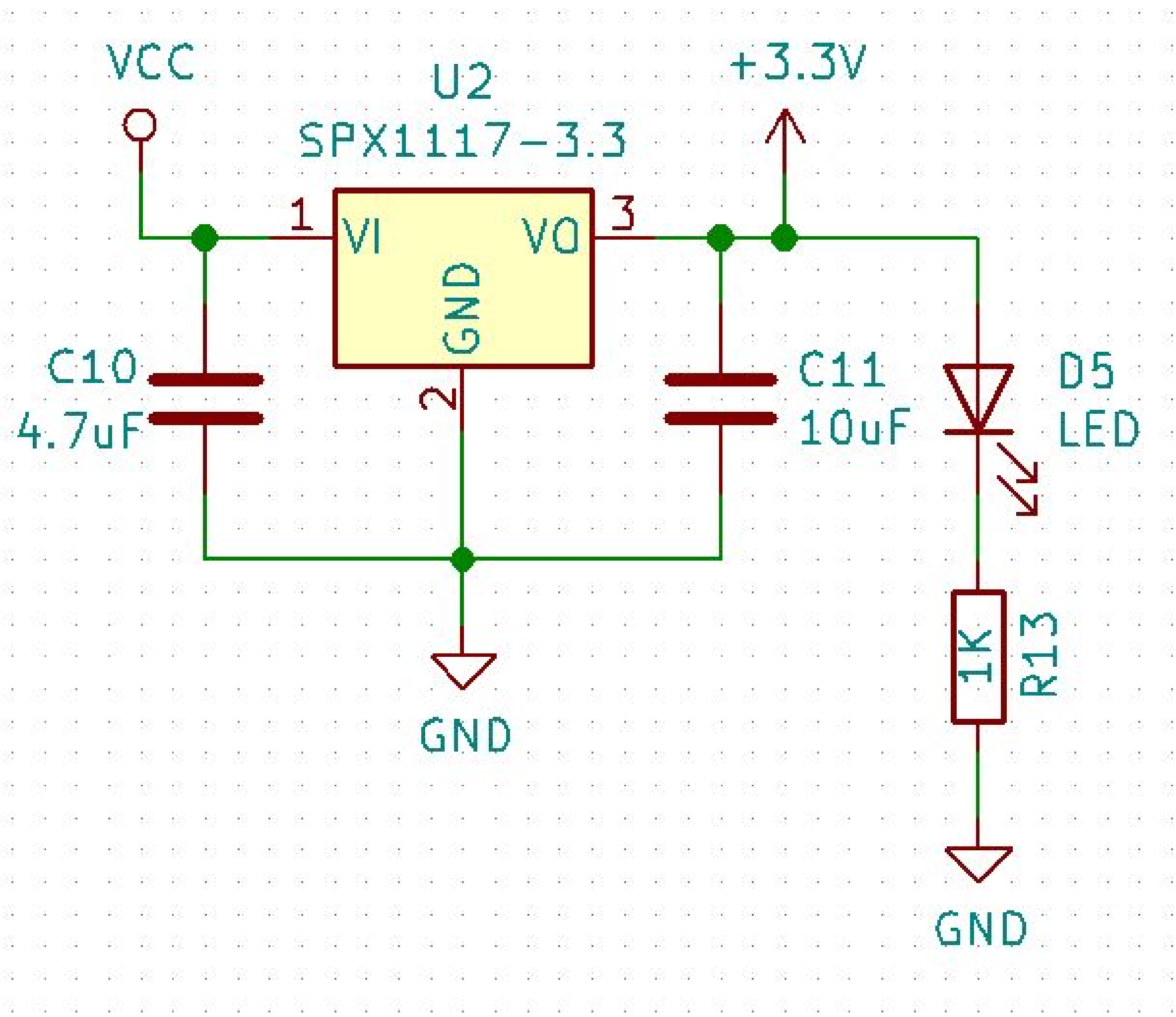
Then, assemble only the switching DCDC as shown below. Select the value of R1 to set whatever you want Vcc to be, in the event you are not using Vcc to power any sensors (all sensors are powered from +24V and/or 3.3V), set Vcc to 5V.
To set Vcc to 5V, R1 should be 316k, for 9V it should be 154k, for 12V it should be 113k, and for 15V it should be 87.7k. Vcc should not be set to 3.3V in typical uses.

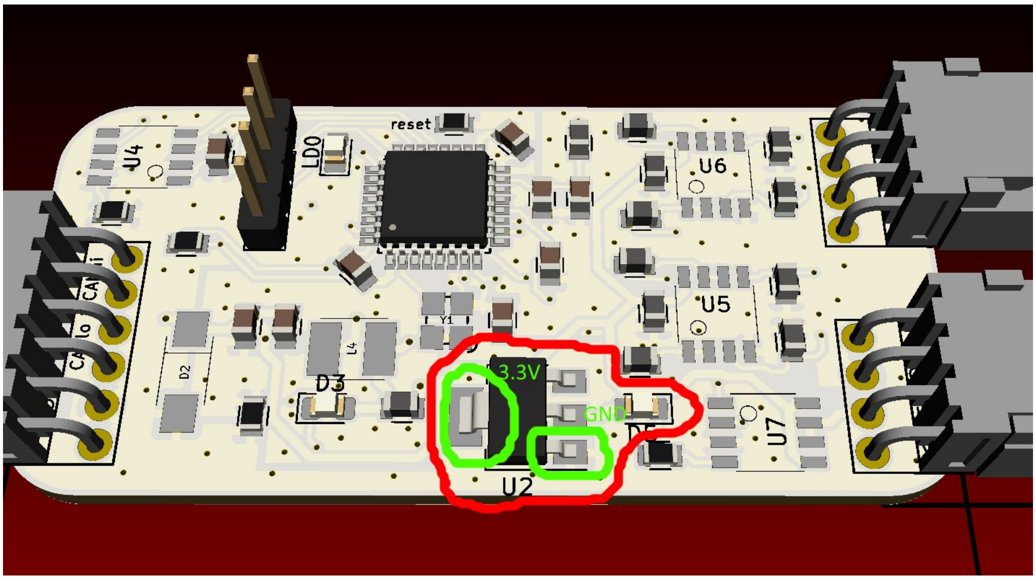


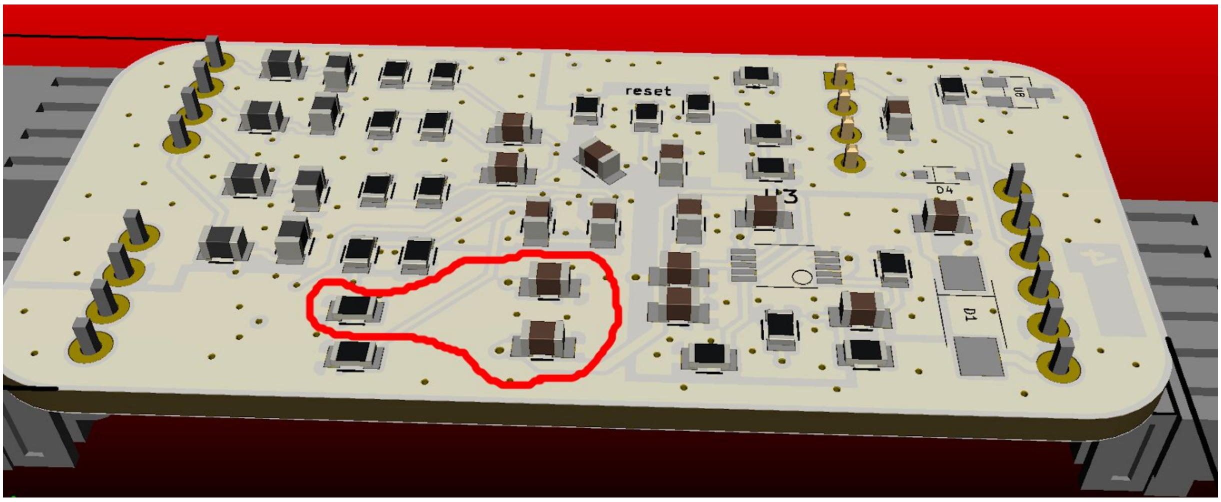


Test points for ground and Vcc are circled in green, you should check that Vcc is the correct voltage by supplying the input (temporarily solder a wire to the circled cyan connector terminals for example) with 24V and checking the output voltage with a multimeter. Additionally, the LED on the DCDC should be illuminated when there is power.
If the DCDC has problems there should be a separate document related to troubleshooting them but that doesn't exist yet or I would direct you to it.

Next, assemble the linear regulator as shown below.


VCC
U2
SPX1117-3.3
+3.3V
1
VI
VO
3
GND
C10
4.7uF
2
C11
10uF
D5
LED
1K
R13
GND
GND





Test points for ground and 3.3V are circled in green, you should check that 3.3V is being supplied to the 3.3V net. Additionally, the LED should be illuminated.

Then, assemble everything else except components labeled DNA. If the node will be used outside housings, wires should be soldered to output pins and connected to an IMC connector, the following pinout is suggested:

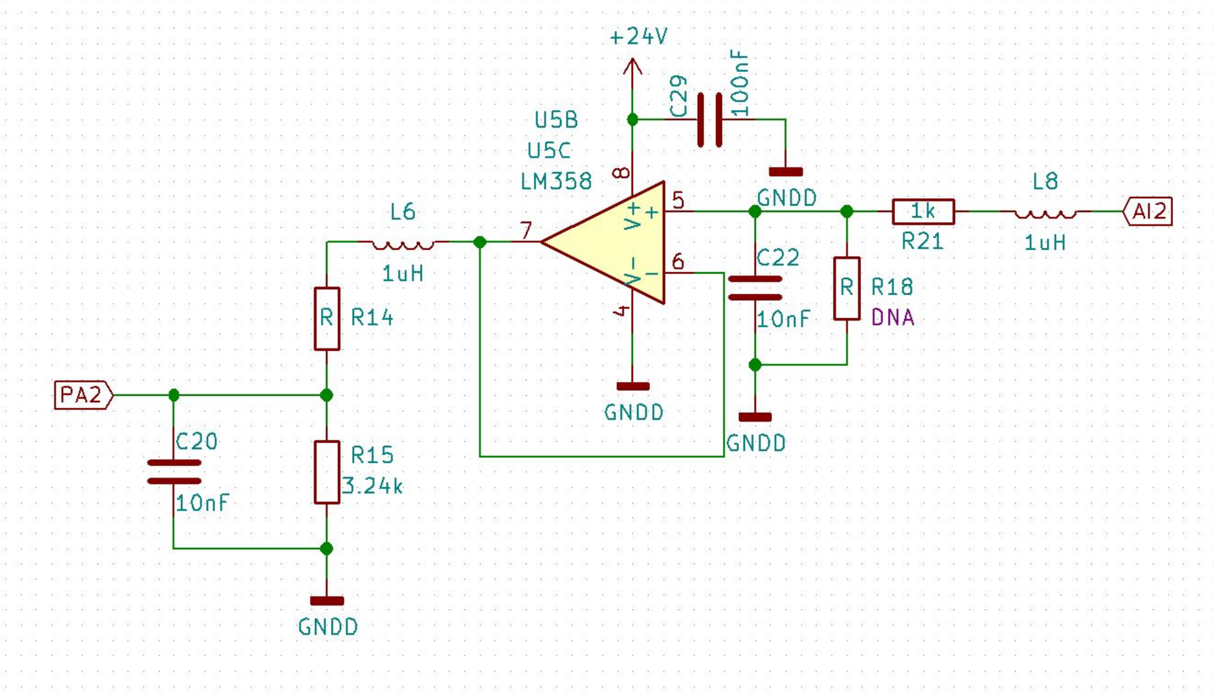
- pin 1: +24V in
- pin 2: CANhi
- pin 3: CANhi
- pin 4: CANlo
- pin 5: CANlo
- pin 6: +24V out OR Vcc out
- pin 7: Vcc out or +3.3V out
- pin 8: AI2
- pin 9: AI3

pin 10: AI5
pin 11: AI6
pin 12: GND

If not all the pins are needed the signal ground (next to sensor power outputs) should be connected to a pin and used as the ground pin for sensors. Additionally, if +24V, Vcc, and +3.3V are all needed an extra connector could be used and signal ground could be put on that connector.

When assembling the voltage dividers, select the value based on the maximum output of the read sensor based on the following table (note that for Vin = 3.3V there is a 1k resistor and the low side resistor is left unassembled, this is to prevent the LC circuit from ringing):

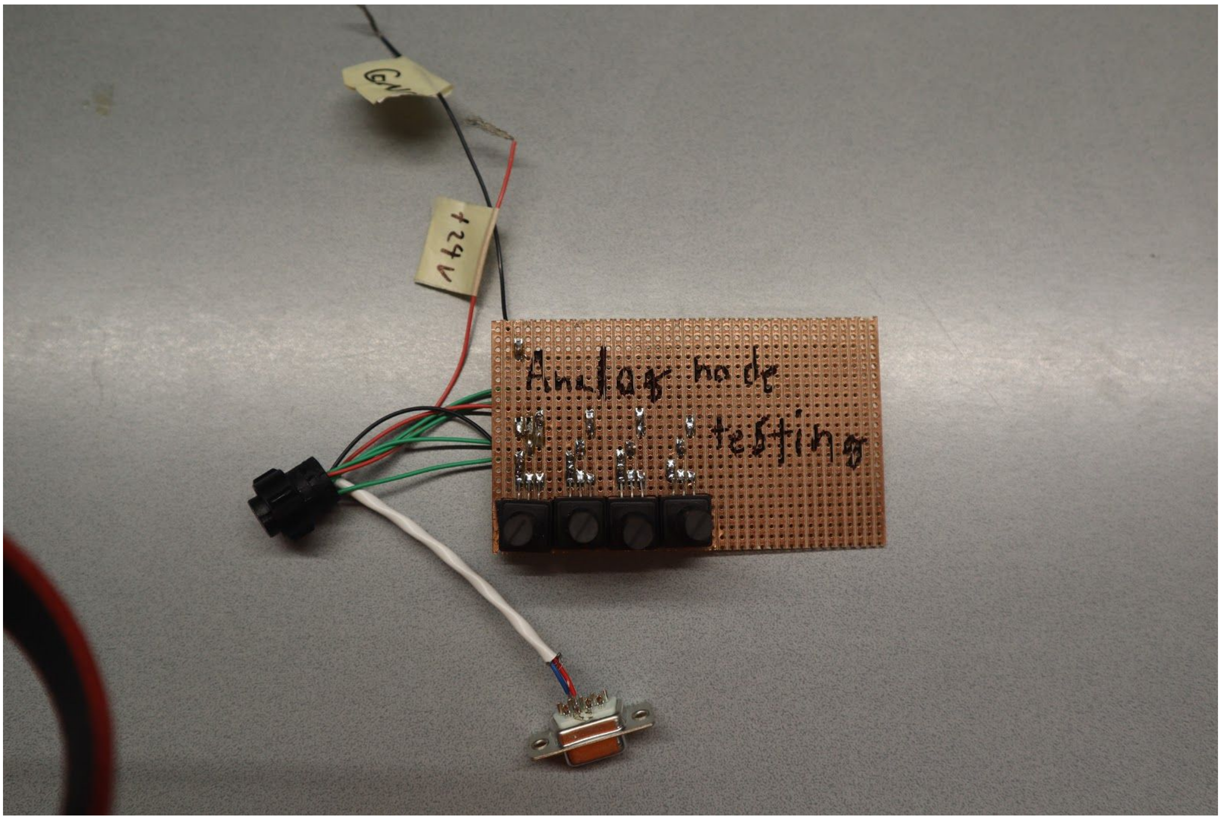
Vin (max)	high side resistor (R14)	low side resistor (R15)
3.3V	1k	DNA
4.5V	1.21k	3.24k
5V	1.74k	3.24k
9V	5.76k	3.24k
12V	8.87k	3.24k



For NTC sensors, the input inductor and 1k resistor should be short circuited and the input low side resistor (R18) should be used as the other side of the voltage divider. The value should be determined based on the NTC resistance and important temperature values. (for the 2020 water/oil temperature sensors, 680 ohm resistors are used)

If a measurement requires more than 11 bit resolution, it is necessary to change the opamp to ADA4522-2. In the case that more than 12 bit resolution is required then the software for hardware oversampling should be implemented, as it has not been tested yet.

Except for output voltages, an assembled board can be tested with the analog node testing board. Configuration of the board should be done first, which will not be discussed here except to say that it should be done in config.c and in order to use the test board inputs should be configured as uncalibrated voltage inputs.



Note that the voltage to the potentiometers is fed from pin 7, so it is at most either 3.3V or Vcc, in the case that it is Vcc it should be ensured that putting Vcc to an input will not break anything (all voltage dividers are intended for input voltages at least at Vcc, or voltage divider is not assembled so input is floating).

The can can be tested at the same time as the analog inputs by connecting the Kvaser and using it to check that the voltages are in a reasonable range when moving the potentiometer. Note that the potentiometers are old, and sometimes there is an internal loss of contact.

This does not test the sensors, so they may have to be tested separately. Similarly, for NTCs it does not test the NTC or low side resistor, but will test the rest of the measurement system.

MCU Pinout

PA2, PA3, PA5, and PA6 are connected to AI2, AI3, AI5, and AI6.

PA9 is the control input for the high side driver for sensors powered from the LV supply voltage.
PA7 is the current / fault monitoring input.
(https://www.st.com/resource/en/datasheet/vn7140aj.pdf)

PB3 is the PG pin of LT3970.
PA0 is from a 3.12V zener, PF1 is from the 3.3V power supply.

PA15 is the debug led.

PA11 is CANrx, PA12 is CANtx.

PF0 is the input from a 16 MHz oscillator that should be used as HSE clock.

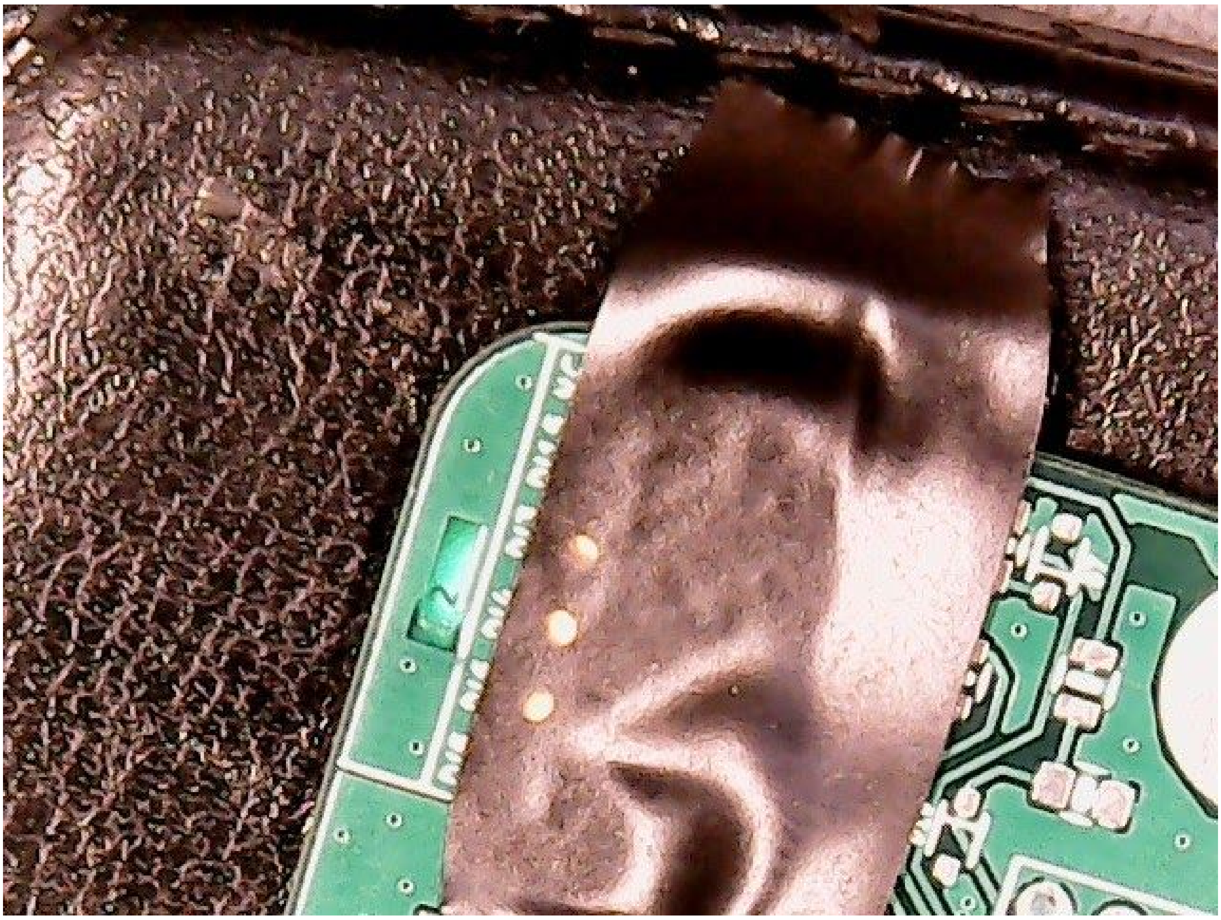
CAN Configuration / Details

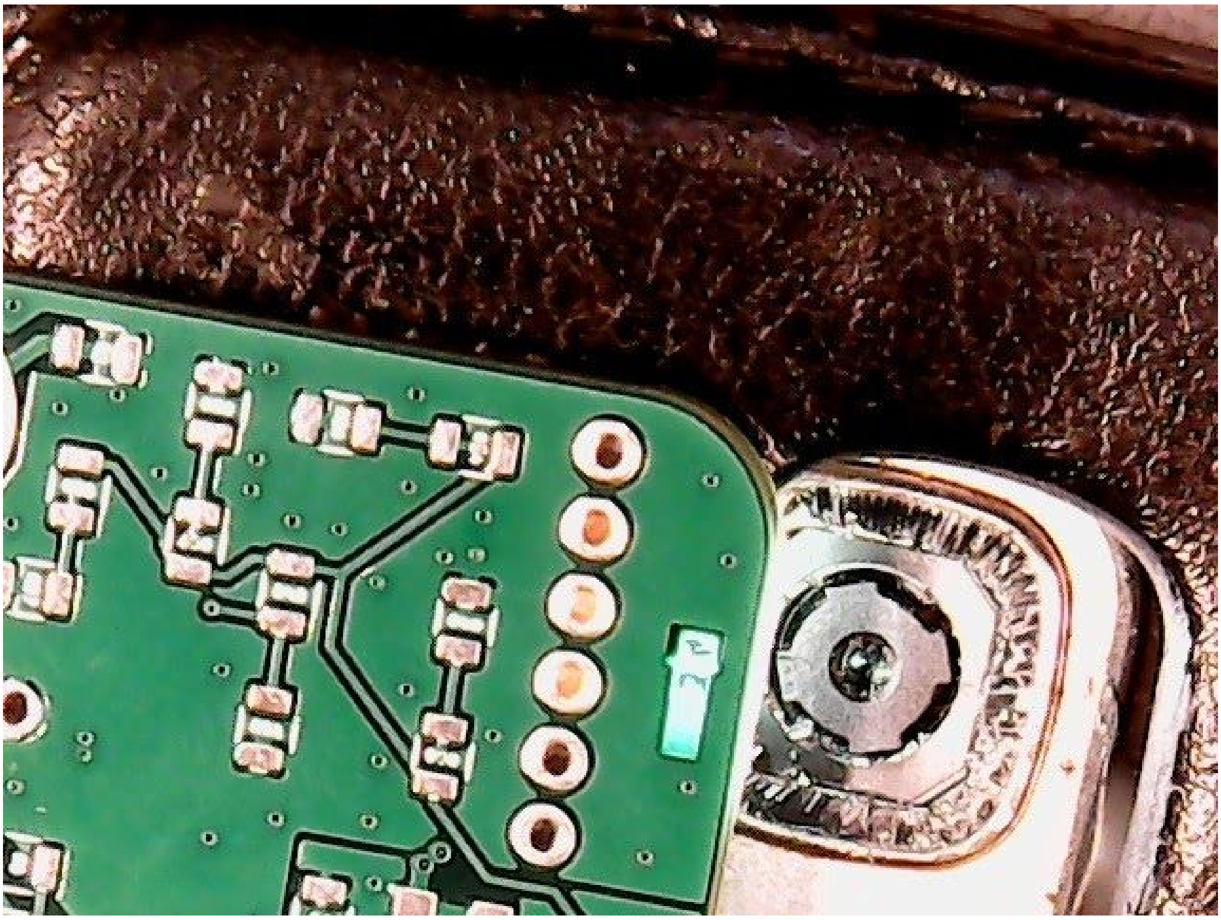
this is currently not implemented

Power Node 2020 Assembly and Programming Guide

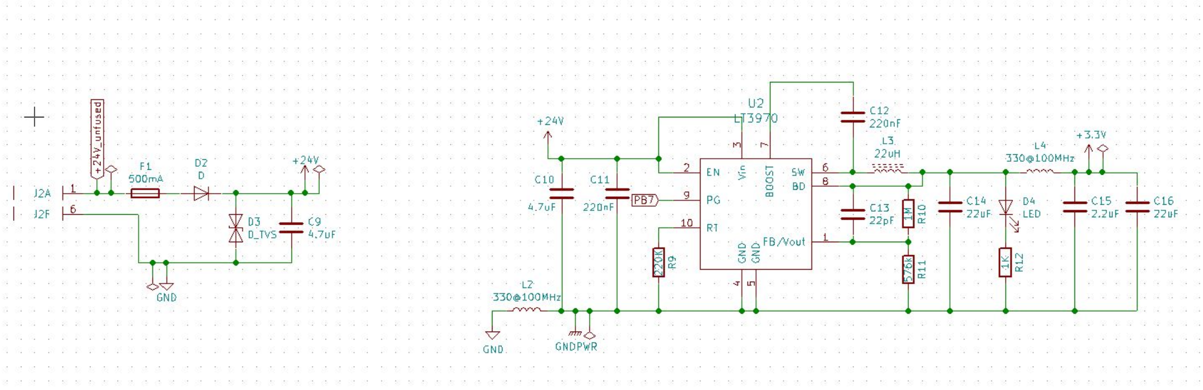
Assembly and Hardware Testing Instructions

First, check that all four layers are present on the board with a light on the place shown in the images.

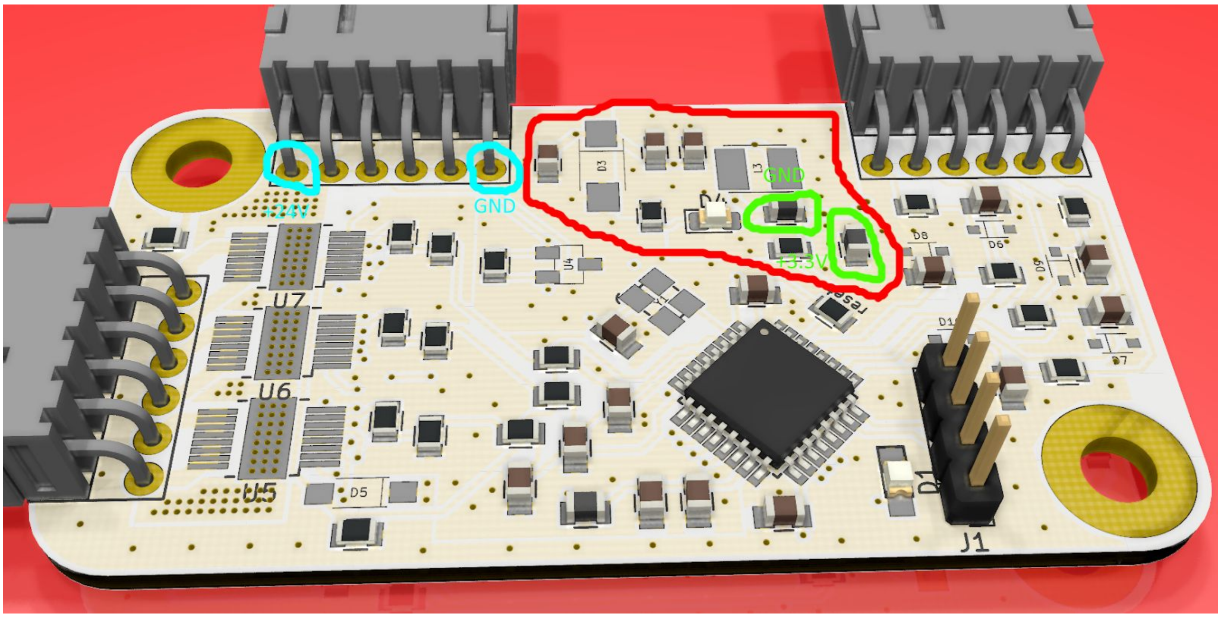


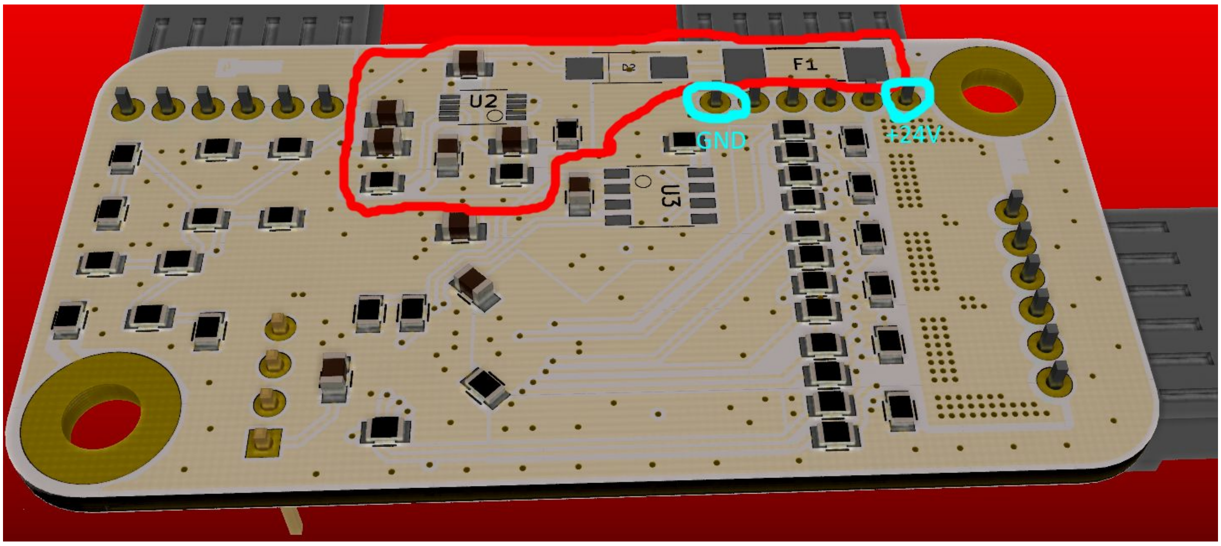


Then, assemble the switching DCDC as shown below.



Metropolia
University of Applied Sciences





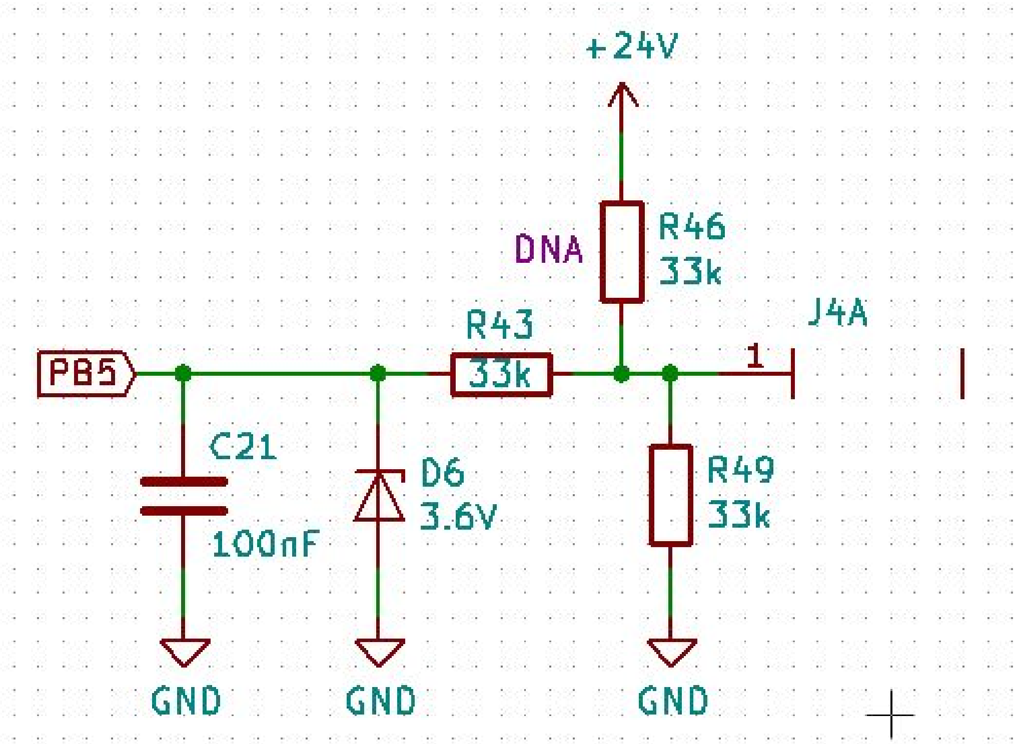
Test points for ground and 3.3V are circled in green, you should check that 3.3V is the correct voltage by supplying the input (temporarily solder a wire to the circled cyan connector terminals for example, or holding probes against the connector terminals) with 24V and checking the output voltage with a multimeter. Additionally, the LED on the DCDC should be illuminated when there is power.
If the DCDC has problems there should be a separate document related to troubleshooting them but that doesn't exist yet or I would direct you to it.

Then, assemble everything else except components labeled DNA. If the node will be used outside housings, wires should be soldered to output pins and connected to an IMC connector, the following pinout is suggested:

pin 1: +24V in

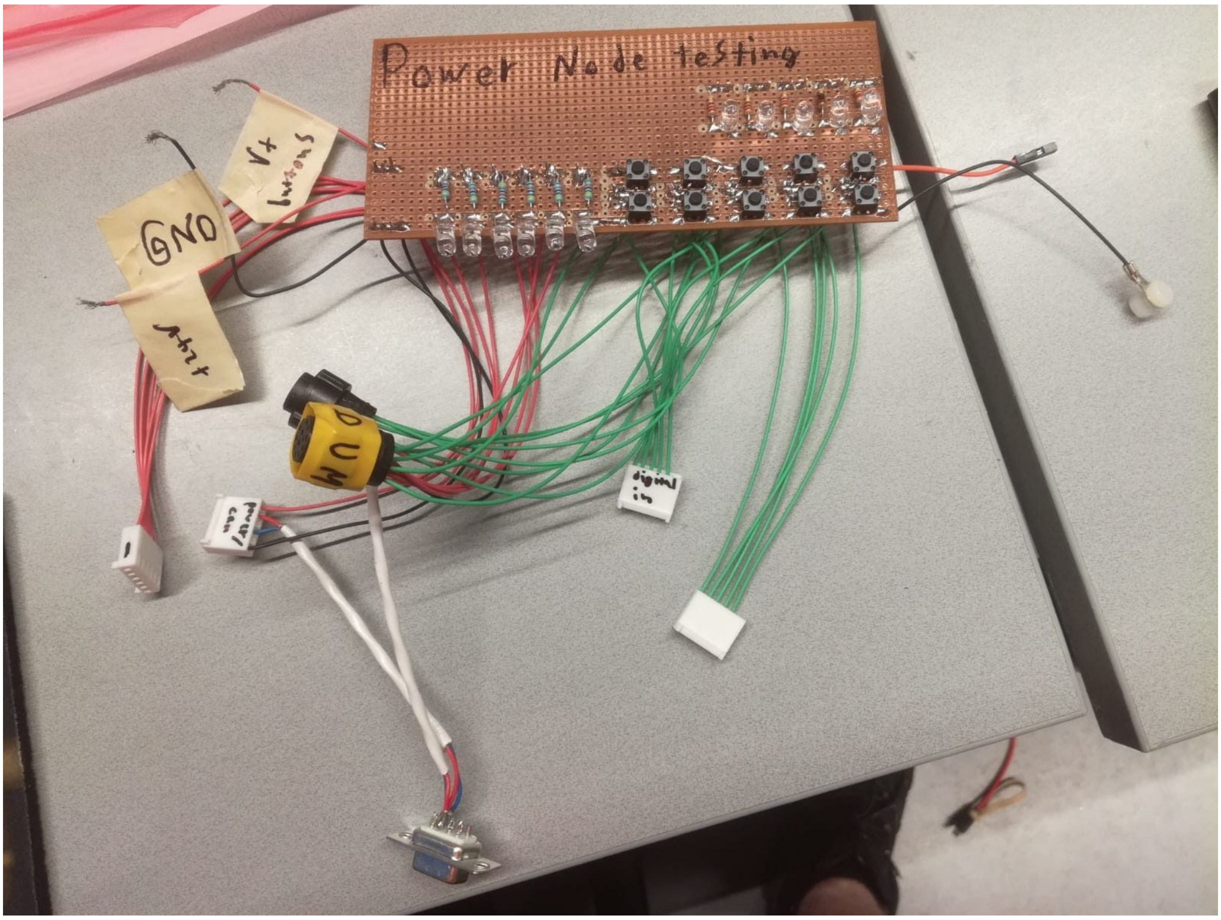
pin 2: CANhi
pin 3: CANhi
pin 4: CANlo
pin 5: CANlo
pin 6: PO5
pin 7: PO4 or DIO15
pin 8: PO3 or DIO3
pin 9: PO2 or ~~DIO4~~ (skip DIO4, PB4 seems to have some problem on the microcontroller)
pin 10: PO1 or DIO6
pin 11: PO0 or DIO5 (note that DIO5 may not read PWM signals, it can only read digital signals or output PWM)
pin 12: GND

When assembling digital inputs the default configuration is digital input with pulldown.



If a pullup is required instead then do not assemble the pulldown resistor (R49) and instead assemble the pullup resistor (R46).
If a pwm output is wanted to control a fan, the series resistor (R43) should be shorted and the rest of the components should not be assembled.

Except for PWM inputs an assembled board can be tested with the power node testing board. Configuration of the board should be done first, which will not be discussed here except to say it should be done in config.c of the power node code.



If connectors are assembled the board can be tested simply by plugging in the appropriate connectors, while if wires are soldered to the board and an IMC connector is used then instead the IMC connector must be configured to have the expected pinout.

Digital inputs can be tested with the buttons, note that the button supply must be connected for any of the high-side buttons to work, and for any pullups on the PCB to work it must have the 24V supply connected. Also note that if a PWM output is assembled, the buttons corresponding to that pin should not be pressed (if connectors are assembled to the board), as it will likely damage the microcontroller.

Power outputs can be tested with the six LEDs on the edge of the board. To test this it is required to have the 24V supply connected.

To test can, connect the a Kvaser (probably with at least one termination resistor) to the DB-9 connector and check with canking if the node is sending. The power node can be temporarily configured to continuously send a message to check this.

It is rare that a PWM input is needed, so it should be tested with whatever PWM output device it will be measuring.

MCU Pinout

PB3, PB4, PB5, PB6, and PA15 are connected to DI3, DI4, DI5, DI6, DI15.
PB4 seems to have some strangle coupling to PA8, so don't use PB4 if it can be avoided.

PA10, PA7, PA6, PA5, PA4, and PA3 are the control inputs for the high side drivers for PO0, PO1, PO2, PO3, PO4, and PO5.
PA2 is current / fault / temperature / voltage monitoring input from PO0 and PO1, PA1 is from PO2 and PO3, and PA0 is from PO4 and PO5.
PA9 and PB0 are SEL0 and SEL1 inputs for all high side drivers to control what the monitoring pin is outputting.
(https://www.st.com/resource/en/datasheet/vn7140aj.pdf)
PF1 is connected to the ground of the high side drivers, as it will have a voltage offset due to the protection diode.

PB7 is the PG pin of LT3970.

PA8 is the debug led.

PA11 is CANrx, PA12 is CANtx.

PF0 is the input from a 16 MHz oscillator that should be used as HSE clock.

CAN Configuration / Details

- 0x80 is sync message, sending it will cause the node to send any messages configured to be sent on sync (for sure not more than 3, and probably not more than 2)

- 0x602 is acknowledge message, after a configuration command it sent it will send the following bytes back:
  [node ID], [command], [ack_counter], [ack_counter], [ack_counter], [ack_counter], [command], [0xFF]
  - ack_counter is a 32 bit uint that increments by 1 every time an acknowledge message is sent

- 0x600 is error or warning message, with the following bytes:
  [node ID], [error_page], [errors_binary], [errors_binary], [errors_binary], [errors_binary]
  - error_binary is a 32 bit number with each bit indicating an error or warning if it is set to 1, the error should be looked up from the main.h of the code
  - error_page is a number of which 32 bit int is being sent, to find the defined error code take the bit number of the error set in error_binary and add 32*error_page to it

- 0x601 is configuration and command message, byte 0 is always node ID and byte 1 is always the command
  - command 1 is used to switch power outputs, byte 2 is the enable byte and byte 3 is the state byte
    - any bits set to 1 in the enable byte mean that the state of the switch is set to whatever that bit of the state byte is, if the bit is set to 0 in the enable byte then that switch is left in its current state

    for example: [3][1][255][255] will close all switches on node 3.

  - command 2 is used to change the duty cycle of PWM outputs, byte 2 is the enable byte and any subsequent byte is the new duty cycle (byte 3 is the duty cycle of the lowest numbered output to be changed, …)
    - enable byte works the same as for command 1, except that instead of a state bit subsequent bytes are used in the order of outputs enabled to be changed

    for example: [7][2][5][255][128] will set the lowest numbered output (DI3) to have a 100% duty cycle and the third output (DI5) to have a 50% duty cycle (if configured as PWM output)

  - commands 3 through 10 are used to request can messages 0 through 7, no additional data bytes are required for example: [2][3] will request message 0 from node 2, and [3][10] will request message 7 from node 3

- command 11 is used to request the microcontroller to takes samples of temperature equal to byte 2 and samples of voltage equal to byte 3

  for example: [7][11][30][10] will request node 2 to take 30 temperature samples and 10 voltage samples

- command 12 is used to clear an error from switch shutoff, until the error is cleared the node will regularly send a warning about which channel has been shutoff, byte 2 should be set to the number of which channel to clear (0 to 5)

  for example: [1][12][3] will clear an error on channel 3 from node 1

- command 128 is save config, it should be used after modifying configuration as otherwise the modifications will be lost when power is cycled

  for example: [12][128] will cause node 12 to save its current configuration variables to flash

- command 129 is used to configure can messages, it uses byte 2 as which message to configure, byte 3 as a command byte, byte 4 as the high byte of the data half word, and byte 5 as the low byte of the data half word
  - if the command byte is from 0-7, it sets the byte length of the corresponding value to be sent

  for example: [1][129][0][0][0][2] will set the length of message 0's first part on node 1 to be 2 bytes and [1][129][0][2][0][1] will set the length of message 0's second part on the same node to be 1 byte

  - if the command byte is from 8-15, it sets the (command byte -8)th value to be sent on that message

    note that if for example the first value to be sent is a 2 byte number, the second value to be sent should be the 3rd byte (command byte 10) and the 2nd byte (command byte 9) is to be skipped; this also applies to the byte lengths (command 2 for the 3rd byte length and command 1 for the second byte length)

    - messages 1 through 6 are current outputs for channels 0 through 5; they are valid for 1 or 2 bytes
    - messages 7 through 9 are temperature outputs for corresponding chips, they are valid for 1 byte

- messages 10 through 12 are voltage outputs for corresponding chips, they are valid for 1 or 2 bytes
- message 13 returns the state of digital inputs on bits 0 to 4
- messages 14, 16, and 18 give pwm input frequency for digital input pins 0, 3, and 4
- messages 15, 17, and 19 are pwm input DC for the same pins
- message 20 gives the states of power output channels on bytes 0 to 5

for example: [1][129][0][8][0][10] will set the message 0's first part on node 1 to be the voltage of U1 (PO0 and PO1) and [1][129][0][10][0][13] will set message 0's second part on node 1 to be the state of all digital IO pins

- if the command byte is 16 the ID of the message is set to the data half word, note that IDs larger than 2^11 are considered disabled

for example: [1][129][0][16][1][1] will set the ID for message 0 on node 1 to be 256 or 0x101

- if the command byte is 17 the DLC of the message is set to the data half word, note that only values 1-8 are valid DLCs

for example: [1][129][0][17][0][3] will set the DLC of message 0 on node 1 to be 3

- command 130 is used to configure the default state of power outputs when the node powers on, it uses byte 2 as the enable byte and byte 3 as the state byte

for example: [17][130][1][255] will set node 17 so that PO0 is closed at powerup

- command 131 is used to configure which can messages will be sent on sync, it uses byte 2 as the enable byte and byte 3 as the state byte

for example: [11][131][3][1] will set node 11 to send message 0 on sync and not send message 1 on sync

- command 132 is used to configure which messages will be sent periodically, it uses byte 2 as the enable byte and byte 3 as the state byte

note that if no messages are enabled to be sent periodically when the node powers on it must be power cycled in order to send messages periodically

for example: [3][132][6][2] will set node 3 to send message 1 regularly and not send message 2 regularly

- command 133 is used to configure how often messages will be sent periodically, it uses byte 2 as the high byte and byte 3 as the low byte of a 16 bit number with 100us resolution

  note that this command is not taken into effect until a power cycle

  for example: [3][133][0][100] will set node 3 to regularly send messages every 10ms

- command 134 is used to configure a delay for sync messages, it uses byte 2 as the high byte and byte 3 as the low byte of a 16 bit number with 10us resolution

  note that this command may not take effect until power cycle, and changing from 0 to any other number or any other number to 0 will cause sync messages to stop working until power cycle

  for example: [8][134][0][100] causes node 8 to have a 1ms delay on sending sync messages

- command 135 is used to configure how often voltage and temperature is read, it uses byte 2 as interval high byte, byte 3 as interval low byte, byte 4 as how many samples of temperature should be taken at each interval, and byte 5 as how many samples of voltage should be taken at each interval
  - the interval has a resolution of 100's of us

  note that if 255 samples are to be taken the power node will instead continuously sample that thing at the same rate as it samples current

  for example: [5][135][0][100][2][1] will make node 5 take 2 samples of temperature and 1 sample of voltage every 10ms

- command 136 is used to configure the default duty cycles for PWM outputs, its bytes are identical to the bytes of command 2

  for example: [2][136][4][128] will set the default duty cycle of the third pin (PB5) as 50% for node 2 on powerup

- command 137 is used to configure the prescalers for PWM outputs (and also inputs), its bytes are identical to the bytes of command 2 except instead of duty cycle prescaler is configured
  - PWM frequency = 666.667/(prescaler+1) KHz (= 170000/((prescaler+1)*255) KHz)

note that for inputs, PWM prescaler determines the minimum readable frequency as 2.5940/(prescaler+1) KHz (= 170000/((presaler+1)*2^16) KHz), but having too high a prescaler will give poor resolution on duty cycle

for example: [1][137][1][32] sets the prescaler for the first PWM pin (PB3) to be 32

- command 138 is used to configure interrupts, byte 2 is which input pin to configure (0-4, but 1 corresponding to PB4 isn't usable), byte 3 is switching power on when detecting a falling edge, byte 4 is switching power on when detecting a rising edge, byte 5 is switching power off when detecting a falling edge, and byte 6 is switching power off when detecting a rising edge
  - if switching off and on are both enabled then power will be switched off on the corresponding edge

- command 139 is used to configure interrupts switching power outputs, byte 2 is which gpio pin to configure (0 to 4), byte 3 enables to switch each power output high on a falling edge, byte 4 enables to switch power output high on a rising edge, byte 5 enables to switch power low on a falling edge, and byte 6 enables to switch power output low on a rising edge

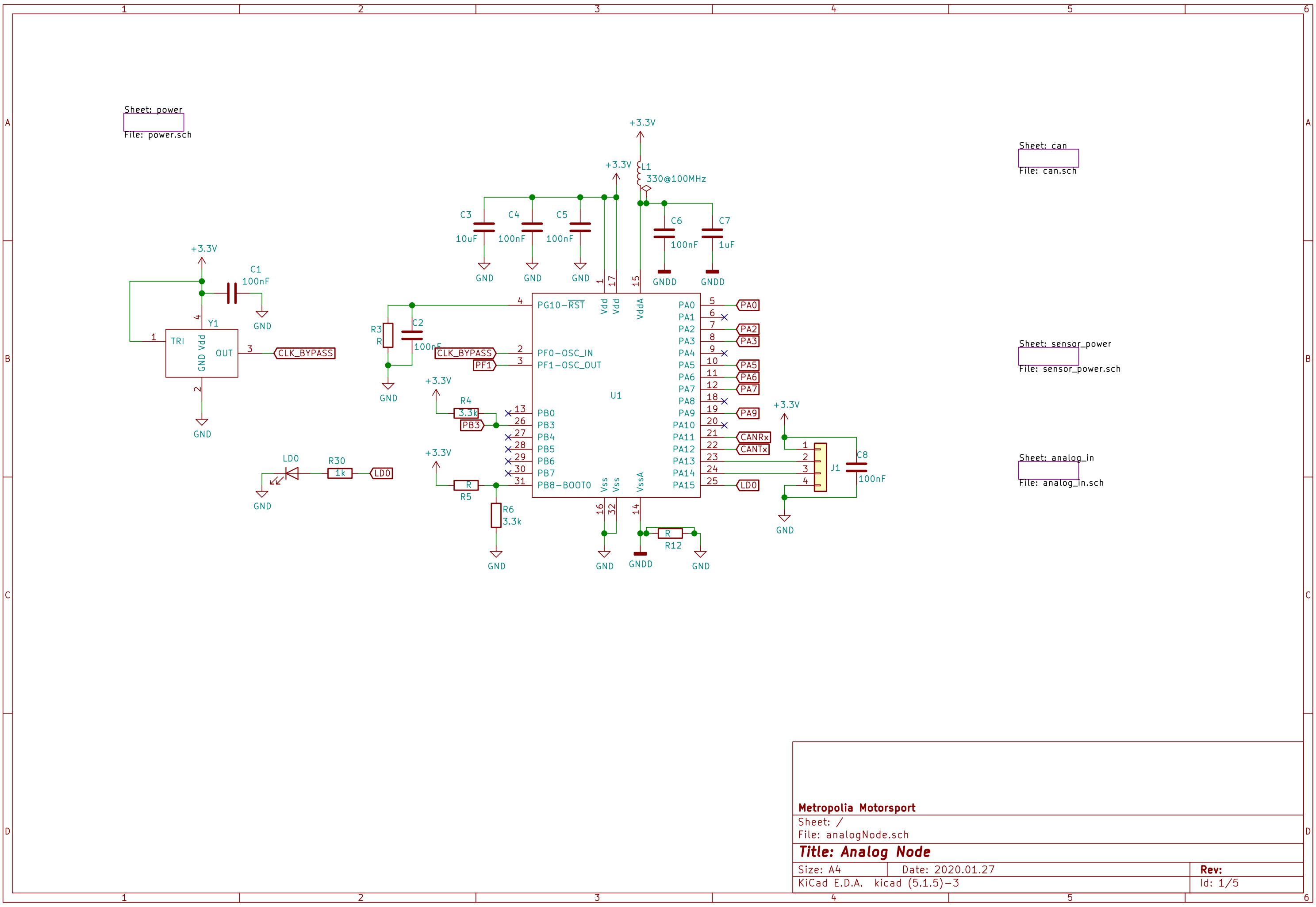
  note that switching low will take precedence over switching high

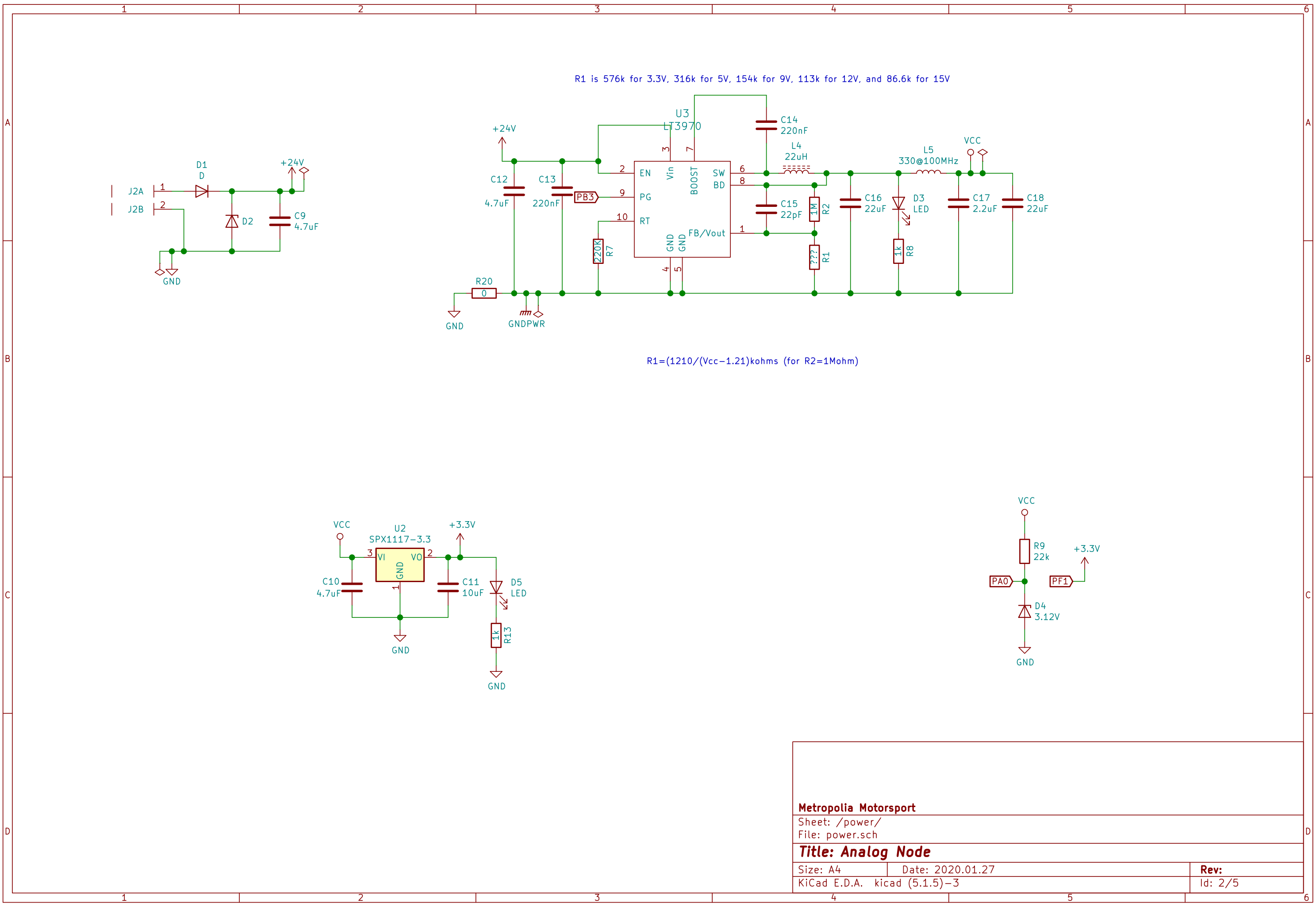
  for example: [8][139][3][1][3][3][1] will cause PO0 to switch off whenever the third digital input (PB5) goes high or low, and PO1 to switch off on a falling edge of the third digital input (PB5), and PO1 to switch on when detecting a rising edge on the third digital input (PB5) if the digital input pin is configured as a digital input

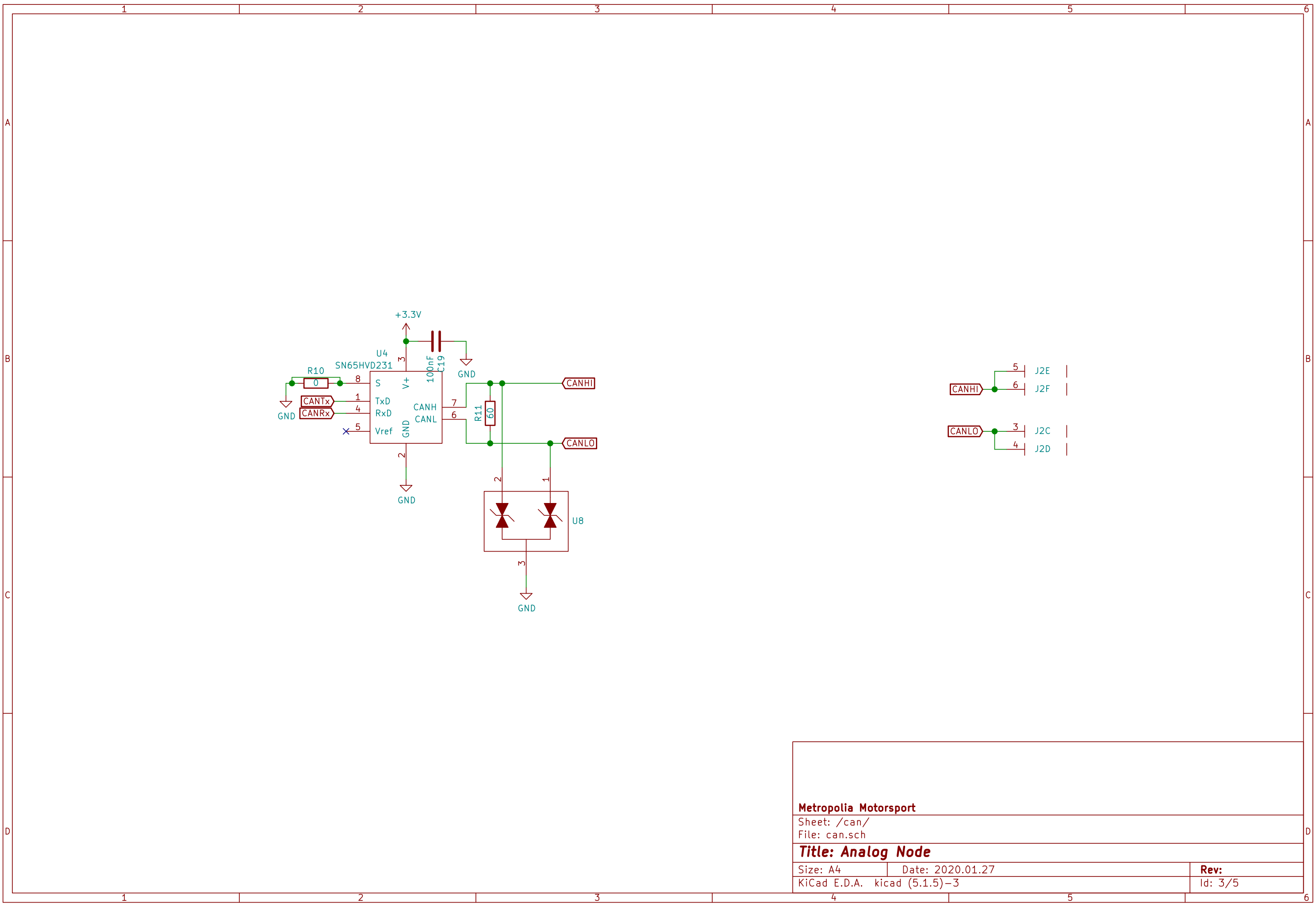
- command 140 is used to configure sending can messages on an interrupt, byte 2 is which input pin to configure (same as command 138), byte 3 is which can message(s) to send on a falling edge, and byte 4 is which can message(s) to send on a rising edge

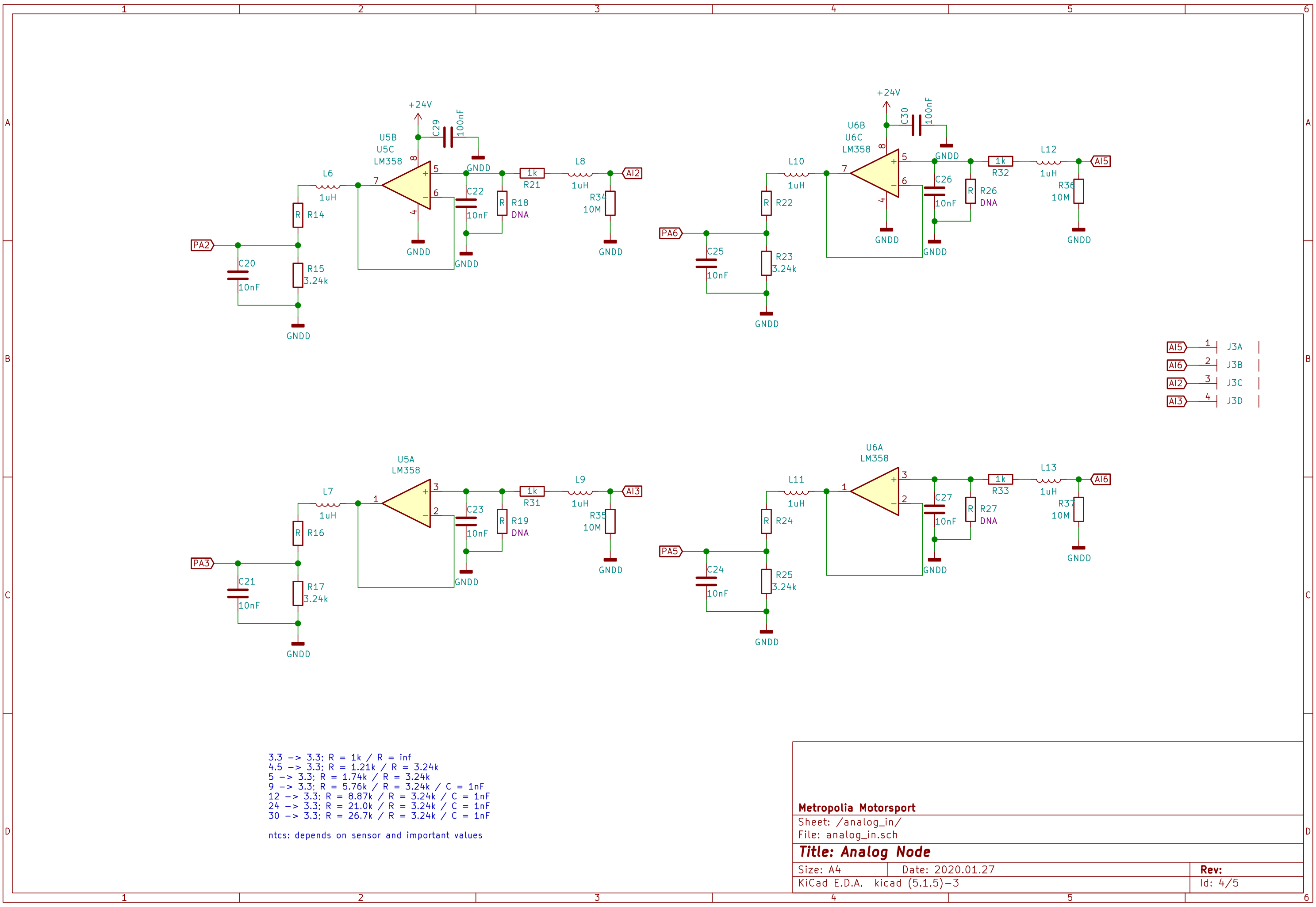
  for example: [1][140][4][3][4] will cause node 1 to send messages 0 and 1 when detecting a falling edge on the fifth digital input (PA15) if it is enabled as a digital input, and to send message 2 when detecting a rising edge on the fifth digital input (PA15) if it is enabled as a digital input

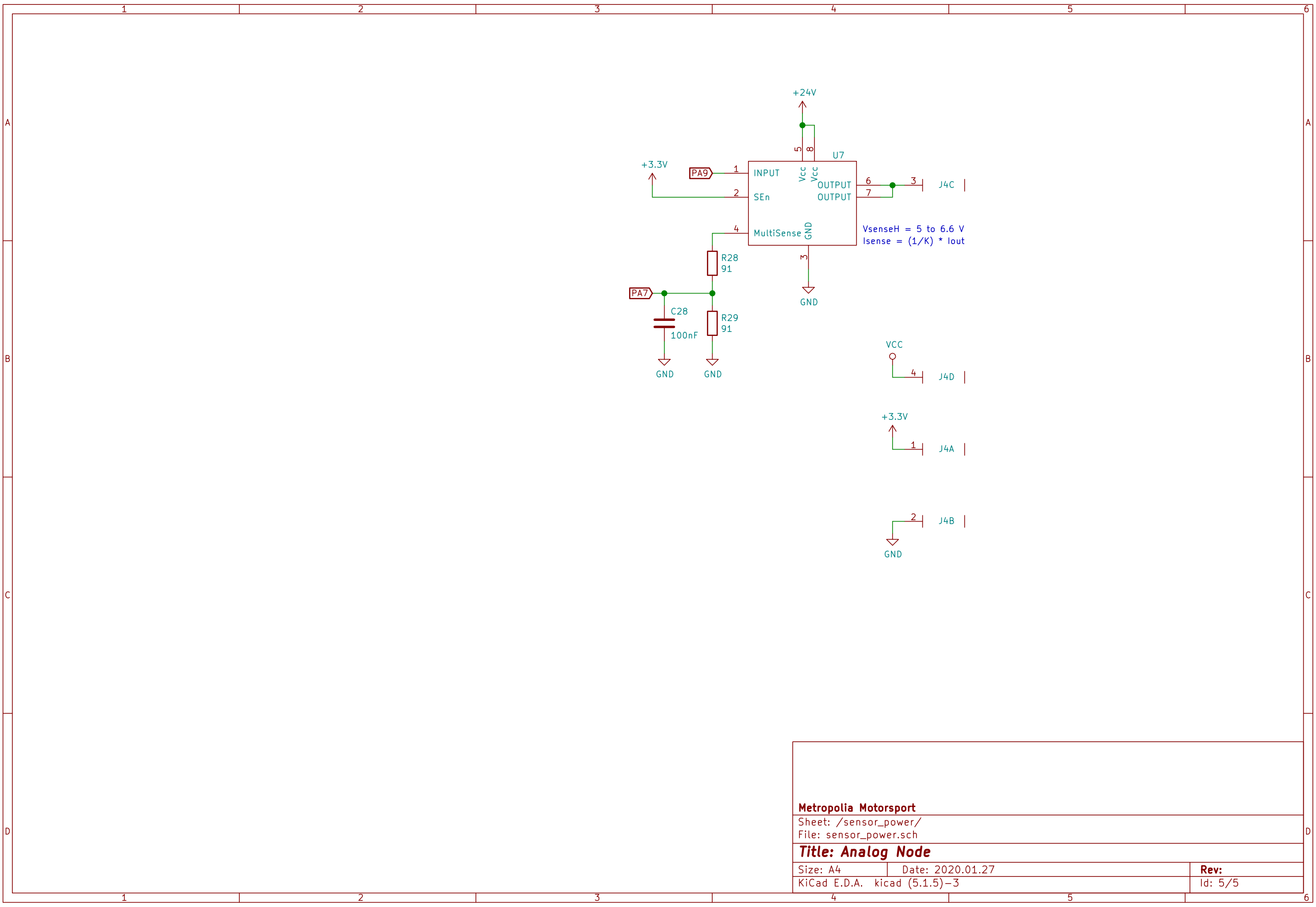
Analog Node Schematic



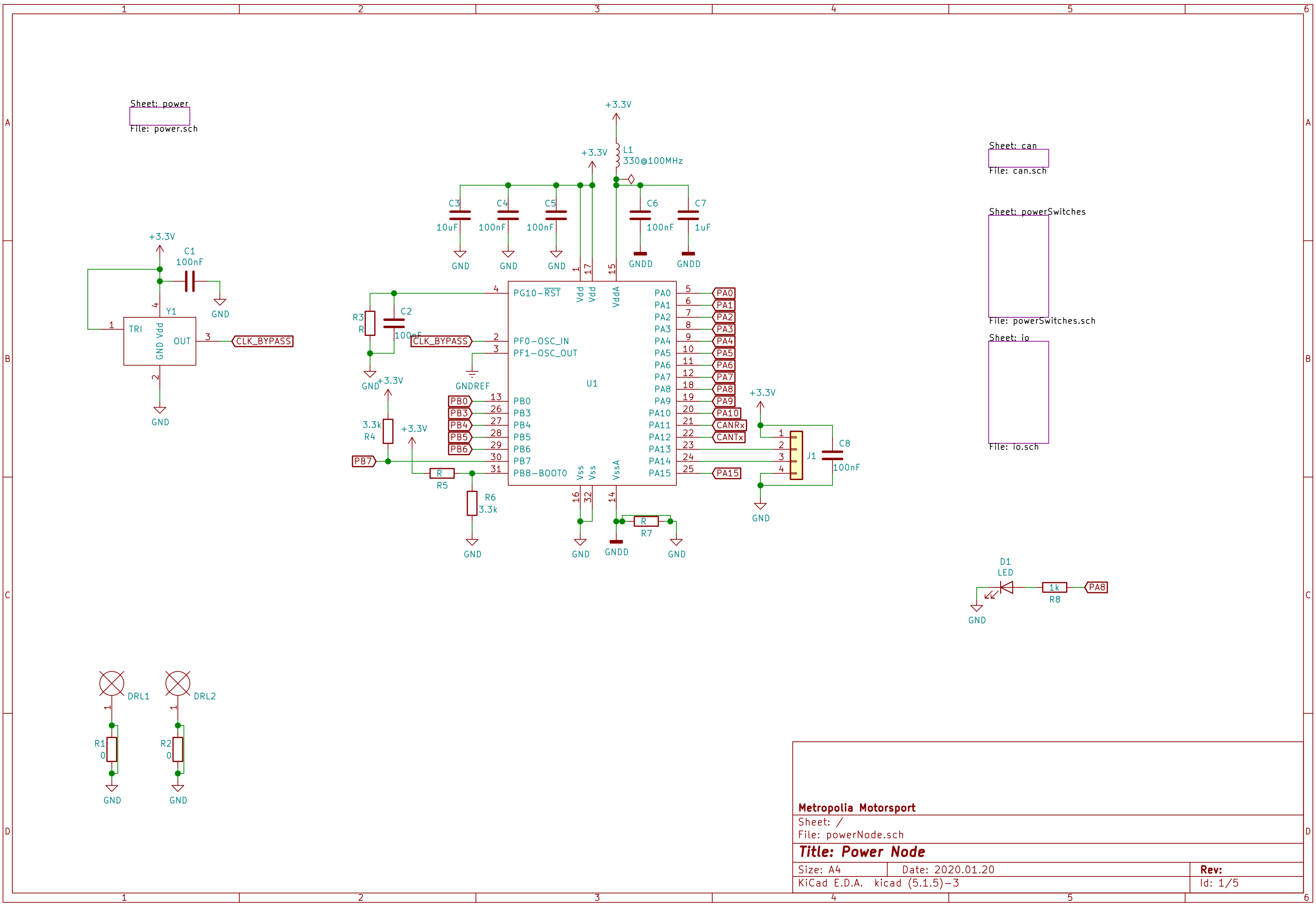

R1 is 576k for 3.3V, 316k for 5V, 154k for 9V, 113k for 12V, and 86.6k for 15V
U3
LT3970
R1=(1210/(Vcc-1.21)kohms (for R2=1Mohm)
U2
SPX1117-3.3
Metropolia Motorsport
Sheet: /power/
File: power.sch
Title: Analog Node
Size: A4
Date: 2020.01.27
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 2/5

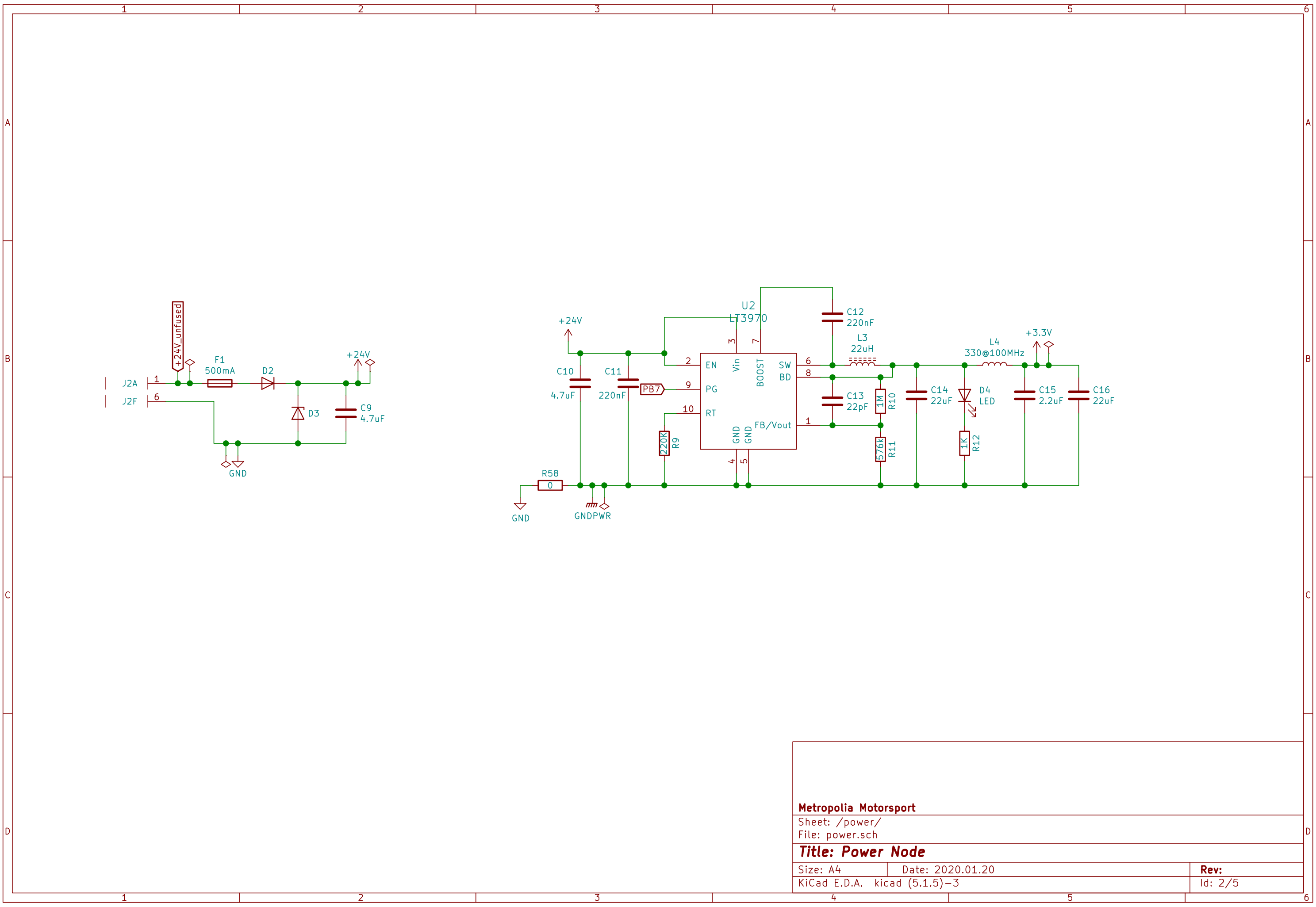

+3.3V
U4
SN65HVD231
R10
0
GND
CANTx
CANRx
S
TxD
RxD
Vref
V+
GND
CANH
CANL
100nF
C19
R11
60
CANHI
CANLO
U8
J2E
J2F
J2C
J2D
Metropolia Motorsport
Sheet: /can/
File: can.sch
Title: Analog Node
Size: A4
Date: 2020.01.27
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 3/5

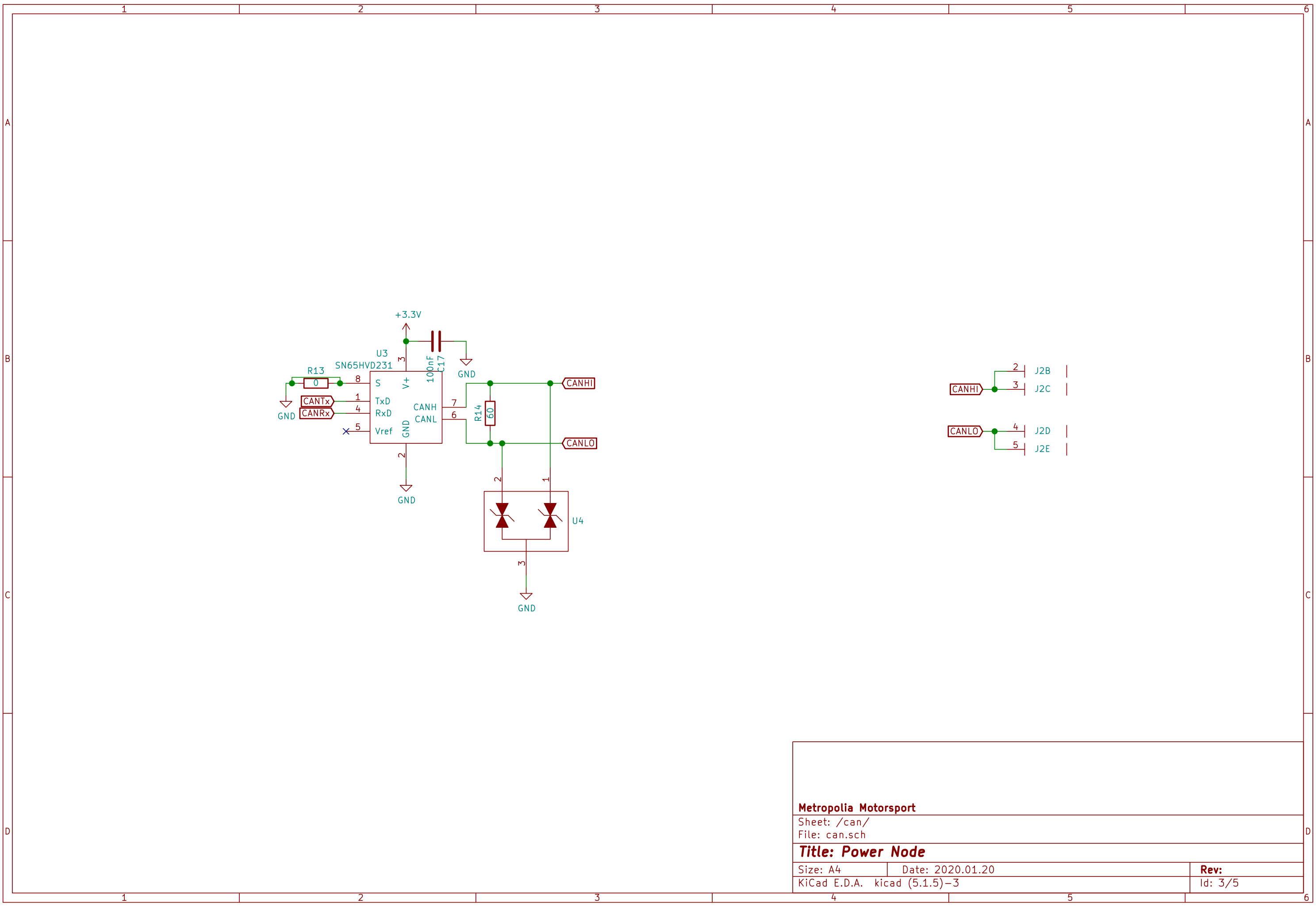

3.3 -> 3.3; R = 1k / R = inf
4.5 -> 3.3; R = 1.21k / R = 3.24k
5 -> 3.3; R = 1.74k / R = 3.24k
9 -> 3.3; R = 5.76k / R = 3.24k / C = 1nF
12 -> 3.3; R = 8.87k / R = 3.24k / C = 1nF
24 -> 3.3; R = 21.0k / R = 3.24k / C = 1nF
30 -> 3.3; R = 26.7k / R = 3.24k / C = 1nF
ntcs: depends on sensor and important values
Metropolia Motorsport
Sheet: /analog_in/
File: analog_in.sch
Title: Analog Node
Size: A4
Date: 2020.01.27
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 4/5

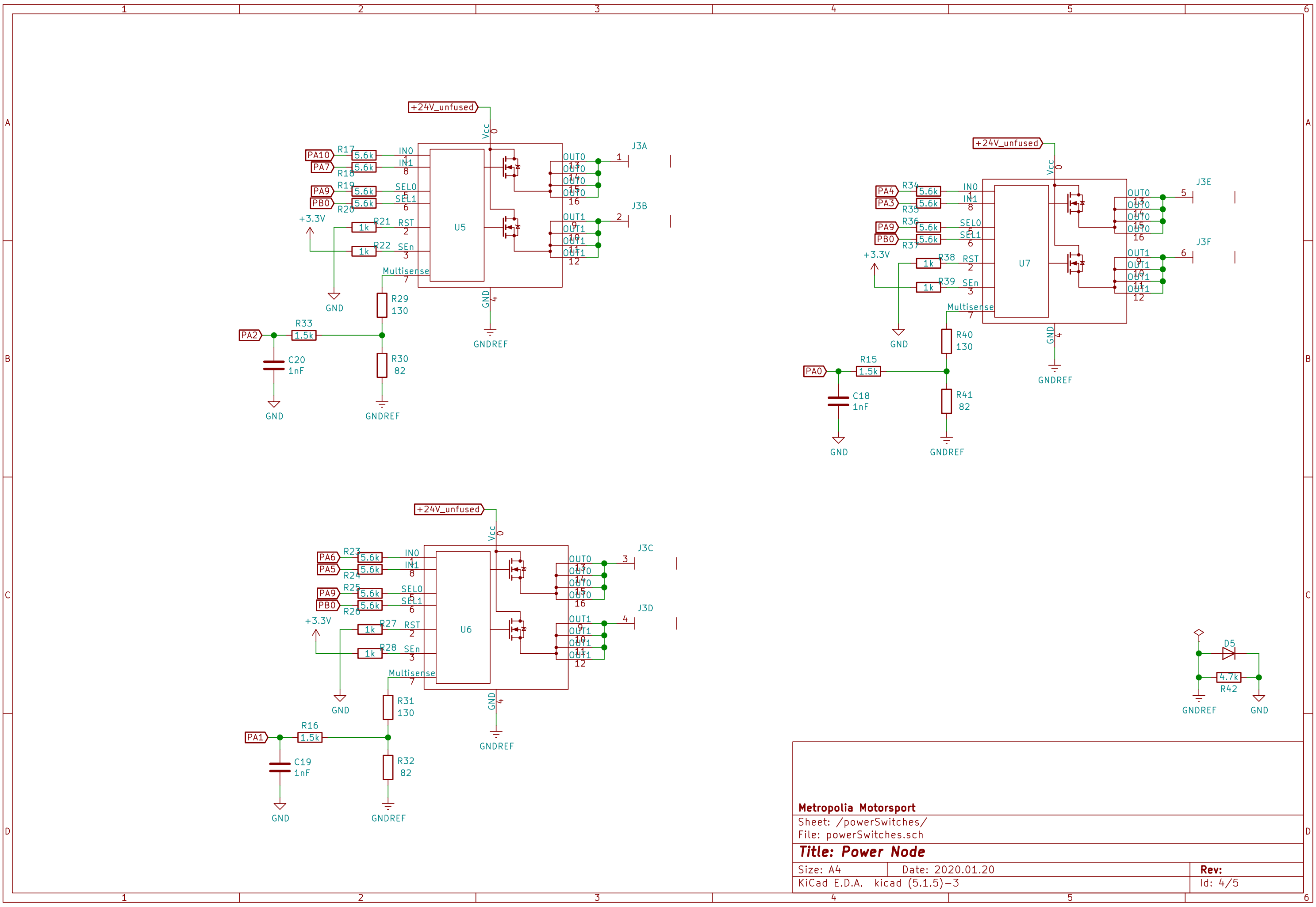

+24V
U7
+3.3V
PA9
INPUT
SEn
Vcc
Vcc
OUTPUT
OUTPUT
J4C
MultiSense
GND
VsenseH = 5 to 6.6 V
Isense = (1/K) * Iout
R28
91
PA7
C28
100nF
R29
91
GND
GND
GND
VCC
J4D
+3.3V
J4A
J4B
GND
Metropolia Motorsport
Sheet: /sensor_power/
File: sensor_power.sch
Title: Analog Node
Size: A4
Date: 2020.01.27
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 5/5

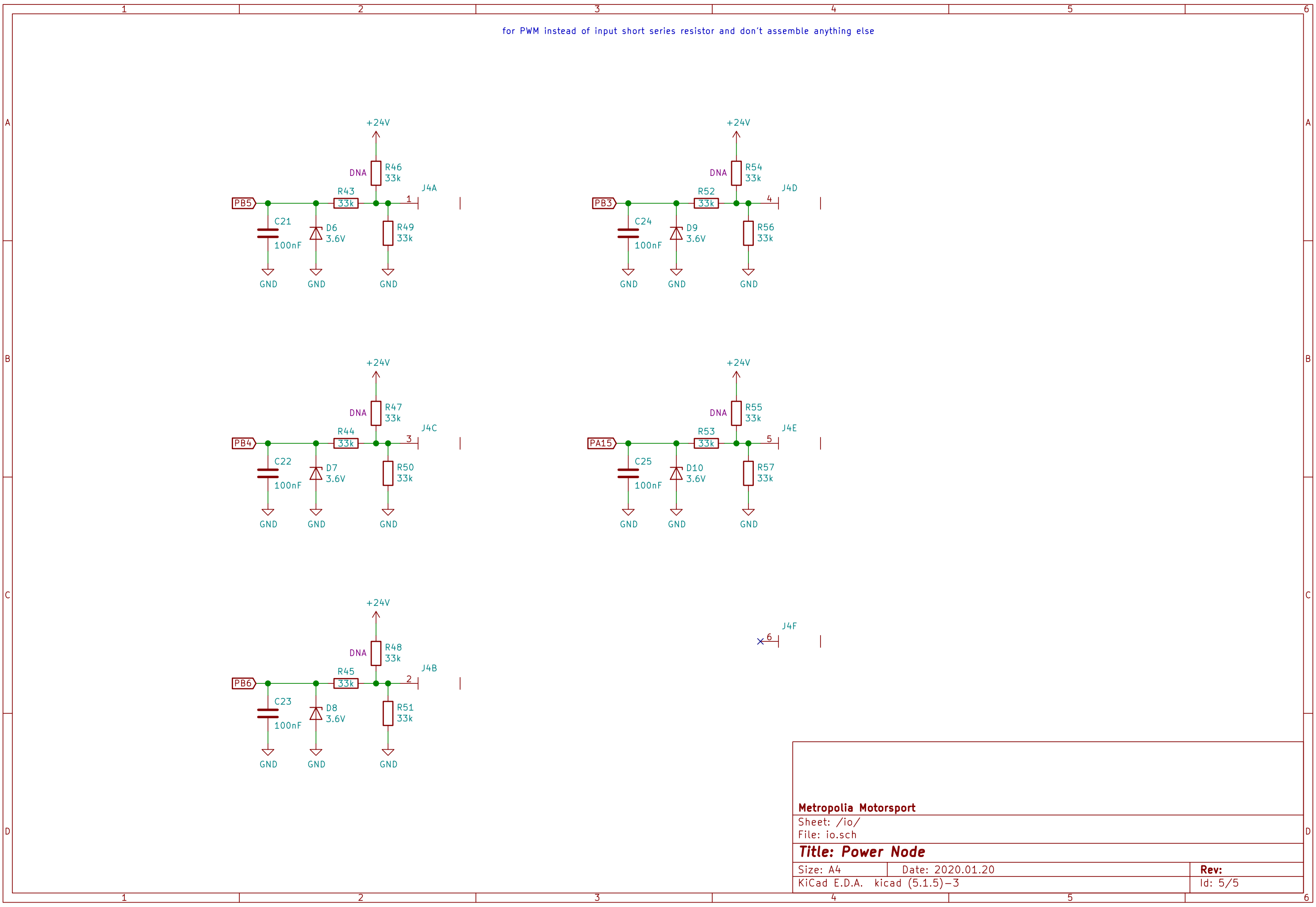
Power Node Schematic




+24V_unfused
J2A
J2F
F1
500mA
D2
D3
+24V
C9
4.7uF
GND
U2
LT3970
+24V
C10
4.7uF
C11
220nF
PB7
EN
PG
RT
Vin
BOOST
SW
BD
FB/Vout
GND
C12
220nF
L3
22uH
C13
22pF
R10
1M
R11
576k
R9
220K
C14
22uF
D4
LED
R12
1K
L4
330@100MHz
+3.3V
C15
2.2uF
C16
22uF
R58
0
GND
GNDPWR
Metropolia Motorsport
Sheet: /power/
File: power.sch
Title: Power Node
Size: A4
Date: 2020.01.20
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 2/5


+3.3V
U3
SN65HVD231
R13
0
GND
CANTx
CANRx
S
TxD
RxD
Vref
V+
GND
CANH
CANL
100nF
C17
GND
R14
60
CANHI
CANLO
U4
GND
GND
J2B
J2C
J2D
J2E
CANHI
CANLO
Metropolia Motorsport
Sheet: /can/
File: can.sch
Title: Power Node
Size: A4
Date: 2020.01.20
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 3/5


Metropolia Motorsport
Sheet: /powerSwitches/
File: powerSwitches.sch
Title: Power Node
Size: A4
Date: 2020.01.20
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 4/5


for PWM instead of input short series resistor and don't assemble anything else
+24V
DNA
R46
33k
R43
33k
J4A
PB5
C21
100nF
D6
3.6V
R49
33k
GND
R54
R52
J4D
PB3
C24
D9
R56
R47
R44
J4C
PB4
C22
D7
R50
R55
R53
J4E
PA15
C25
D10
R57
R48
R45
J4B
PB6
C23
D8
R51
J4F
Metropolia Motorsport
Sheet: /io/
File: io.sch
Title: Power Node
Size: A4
Date: 2020.01.20
Rev:
KiCad E.D.A. kicad (5.1.5)-3
Id: 5/5