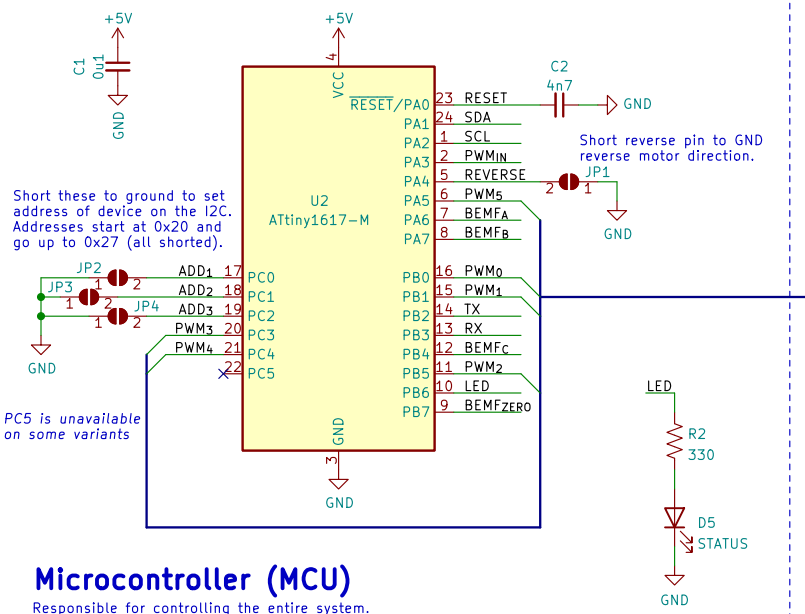


Microcontroller (MCU)

Responsible for controlling the entire system.
Recieves input and adjusts the motor speed accordingly.
Can also output current motor performance using digital communication.

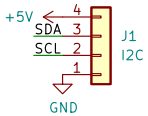
Internal comparator is used for BEMF "zero"-cross detection.
A timer is used to create a time delayed event from the cross event
based on the time recorded between the cross and previous commutation.

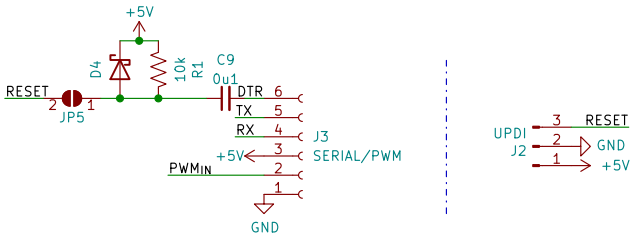
Has an LED to provide feedback to the user.

Control Connectors

Either I2C or PWM can be used to control this ESC.
I2C can act as both input and output for digital data.

PWM is standard for commercial ESCs. 1 to 2ms duty,
20ms period. PWM input is on the UART programming header.

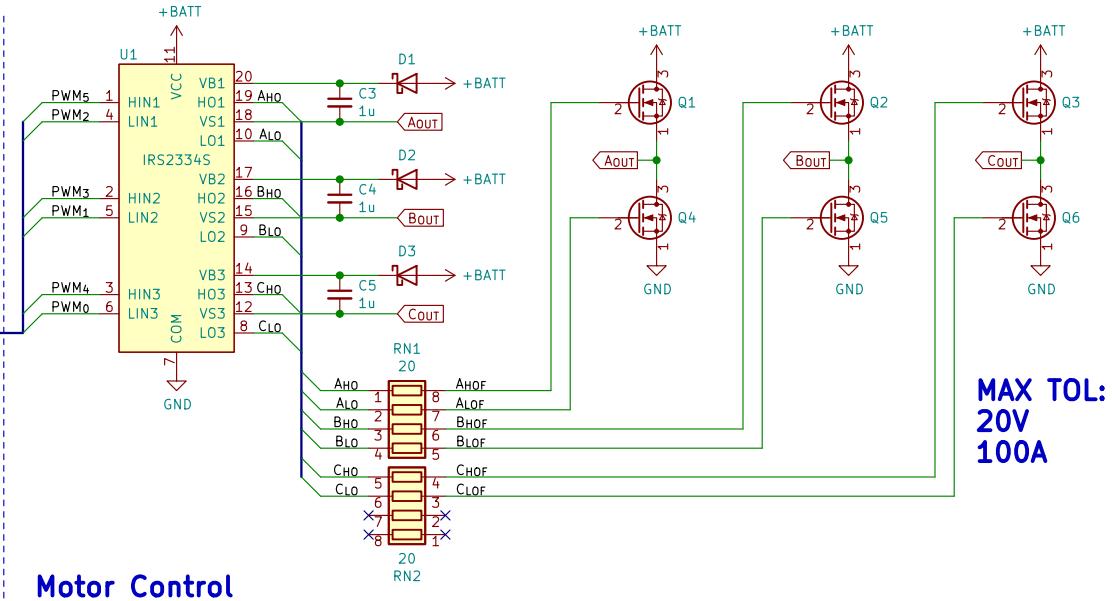




Programming Headers

The MCU is programmed by default using UPDI, but a bootloader can be
programmed so that it can be programmed like other Arduinos using UART.

There are connectors for both of these.



Motor Control

A MOSFET driver is used to drive the half-bridges for each phase of the motor based on control signals from the MCU.

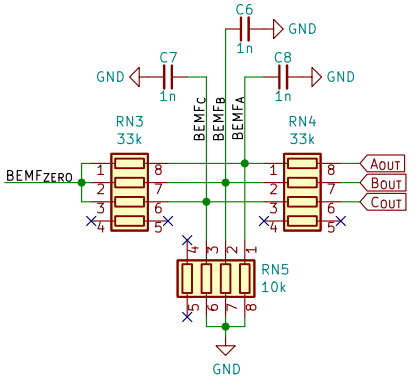
N-channel MOSFETs are used on both sides of the bridge due to their lower "on" resistance. To drive the high-side FETs requires "bootstrapping" to get a gate voltage higher than the drain (VBAT) for optimal performance.

Although the MOSFETs are rated for >100A, the traces are not! Add solder to increase current capacity

BEMF Dividers

Back ElectroMotive Force (BEMF) is generated by a rapidly spinning motor. We monitor this on the floating phase to determine when it is the right time to commute the motor to the next step.

The resistor network is used to divide the voltage down to a safe level (<5V) and derive a virtual zero by averaging the voltage from each phase. These resistor networks need tight tolerances so the division ratios are as identical as possible. Capacitors are there to help smooth out the signal, but are entirely optional.



Motor Control Therory

The basic therory for controlling a 3-phase BLDC motor is that when two phases are connected to high and low, then the third floats [between them].

Let's say phase A is connected high, B is low, and C is left floating (not connected to either power level) - but was previously high.

The floating voltage starts close the previous extreme it was connected to, (high for C) then as the motor rotates through the step it will be pulled toward the other extreme in a roughly linear fashion.

When this floating voltage crosses the average of all three phases ("zero"), we are halfway through a commutation step, so we wait an equivalent amount of time from the previous commutation to the zero crossing before going to the next step. This is all handled by the MCU and its peripherals.

ATtiny1617 based BEMF ESC for BLDC
Designed for 10 to 20V input, current up to 100A
Controlled primarily by I2C but can run with PWM
Same as V4 but with different MOSFET driver and status LED

Title: Sensorless BLDC ESC V5

Savo Bajic	Date: 2022-01-20	**Rev: 5**
KiCad E.D.A. kicad 6.0.1-79c1e3a40b-116-ubuntu21.04.1	Size: USLetter	Sheet: 1/1