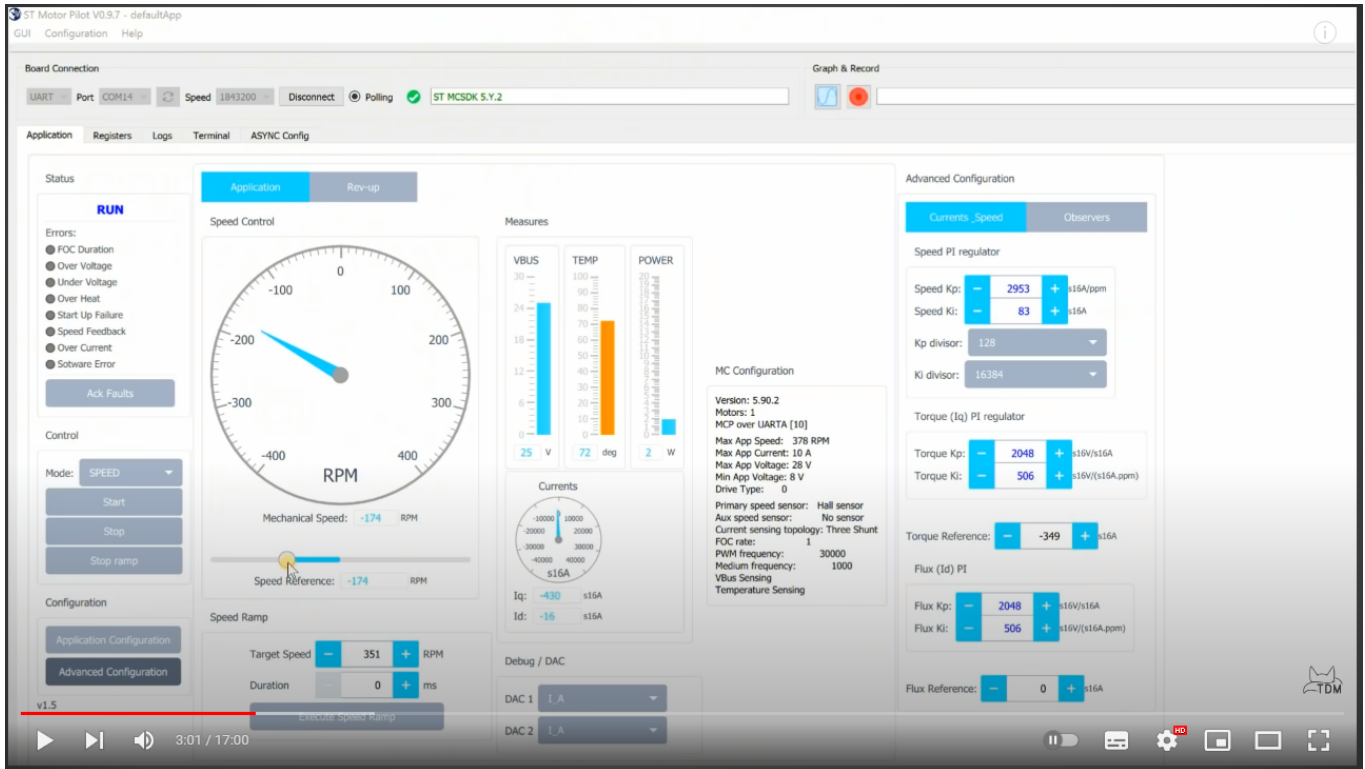
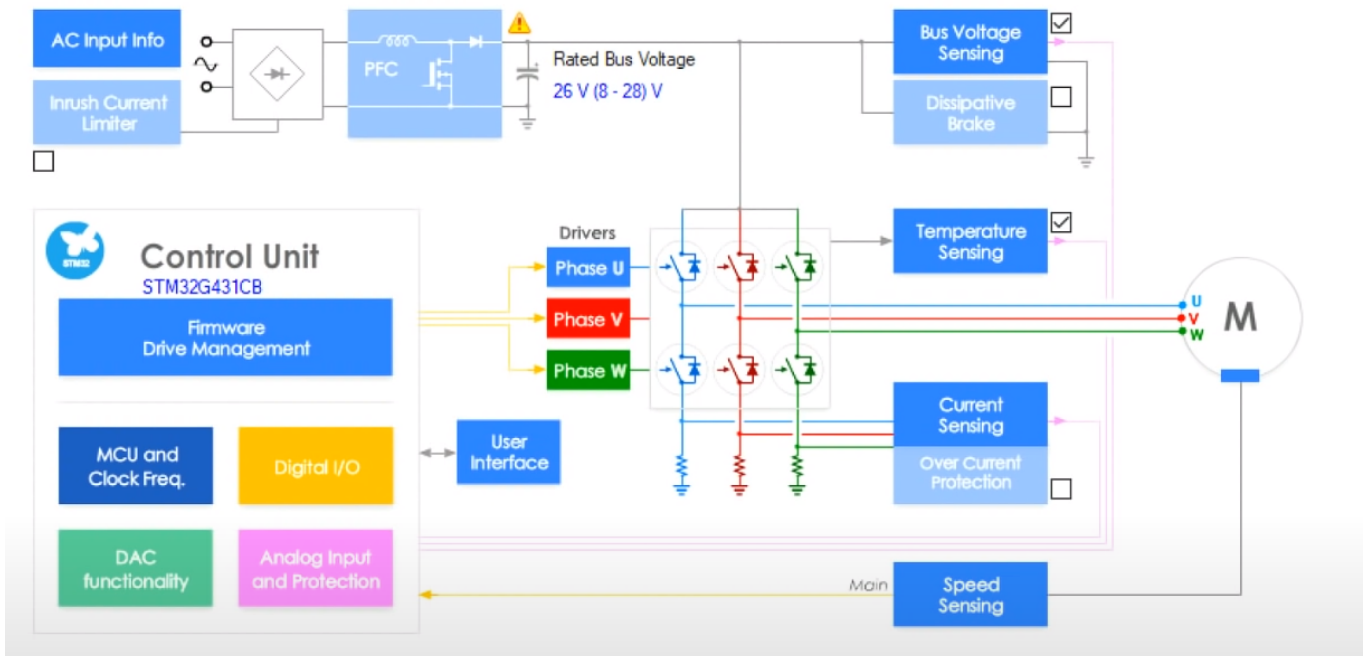


инфу брал отсюда <https://www.youtube.com/watch?v=8V1rTIZczZw&t=369s>

Для отладки и настройки драйверов юзают STM32 Motor SDK.
Крутая штука



<https://www.st.com/en/embedded-software/x-cube-mcsdk.html>



тут еще приведен офигенный материал для подбора компонентов и принципа управления BLDC драйверами

<https://www.ti.com/seclit/ml/sszp343/sszp343.pdf>

ОЧЕНЬ СОВРЕМЕННО использовать готовые трехфазные драйверы которые имеют возможность подключения к МК посредством цифрового интерфейса

DRV8320  ACTIVE

65-V max 3-phase smart gate driver

Order now

[DATA SHEET](#)  [DRV832x 6 to 60-V Three-Phase Smart Gate Driver datasheet \(Rev. C\)](#) | [Online data sheet](#)

[Product details](#) | [Technical documentation](#) | [Design & development](#) | [Ordering & quality](#) | [Support & training](#)

Product details

[Parameters](#) | [Package | Pins | Size](#) | [Features](#) | [Description](#)

Rating	Catalog
Control method	External Control, Trapezoidal Control
Architecture	Gate Driver
Control interface	6xPWM, 3xPWM, 1xPWM
Gate drive (A)	1



<https://www.ti.com/product/DRV8320>

[https://www.ti.com/lit/ds/symlink/drv8320.pdf?](https://www.ti.com/lit/ds/symlink/drv8320.pdf?ts=1691401320749&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV8320)

[ts=1691401320749&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV8320](https://www.ti.com/lit/ds/symlink/drv8320.pdf?ts=1691401320749&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV8320)

Далее важным компонентом будут транзисторы мосфеты. Их уже надо подбирать самим, но самый важный параметр - низкая емкость затвора, чтобы можно было чаще переключать транзистор

Остальные комплектующие подбираются из документа выше

<https://www.ti.com/seclit/ml/sszp343/sszp343.pdf>

Кстати, я дурак. Есть готовая утилита от ST которая сама делает модель двигателя для управления

B-G431B-ESC1
STM32G431CB



Single Motor
ST-LINK/V2 Embedded

[Product Web Page](#)

B-G431B-ESC1 3Sh
STL180N6F7,L6387



Bus Voltage: 10 - 28 Vdc
Output peak current: 1 - 40 A

[Product Web Page](#)

Pole Pairs: [How to detect...](#)

Speed and Current limits

Max Speed: RPM


Max Current: Apk 1 - 40 Apk

VBus: V 10 - 28 V

Magnetic: ☒ SM-PMSM ☐ I-PMSM

 Remember to properly configure the boards in Motor Control mode

 Connect...

 Start Profile

 Save...

 Play

Electrical Model



Mechanical Model




после калибровки получаем такую штуку

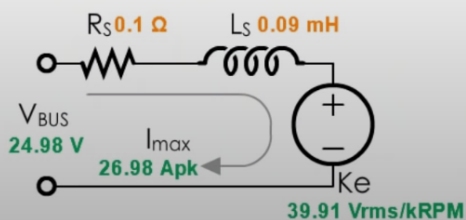
 Disconnect

 Start Profile

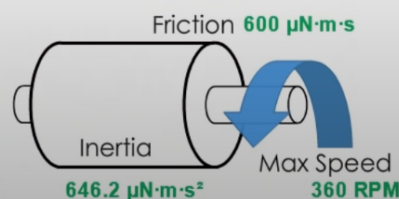
 Save...

 Play

 Electrical Model



Mechanical Model



и далее эти параметры просто подгружаются в motor sdk

Motor - Parameters

Motor Sensors

Magnetic structure Surface Mounted PMSM

Electrical parameters

Pole Pairs	20	
Max. Application Speed	378	rpm
Nominal Current	15.00	Apk
Nominal DC Voltage	26.2	V
Rs	0.10	Ohm
Ls	0.108	mH
B-Emf constant	42.2	Vms/krpm
Inertia	505.915	uN*m*s2
Friction	110.946	uN*m*s

Save parameters Done

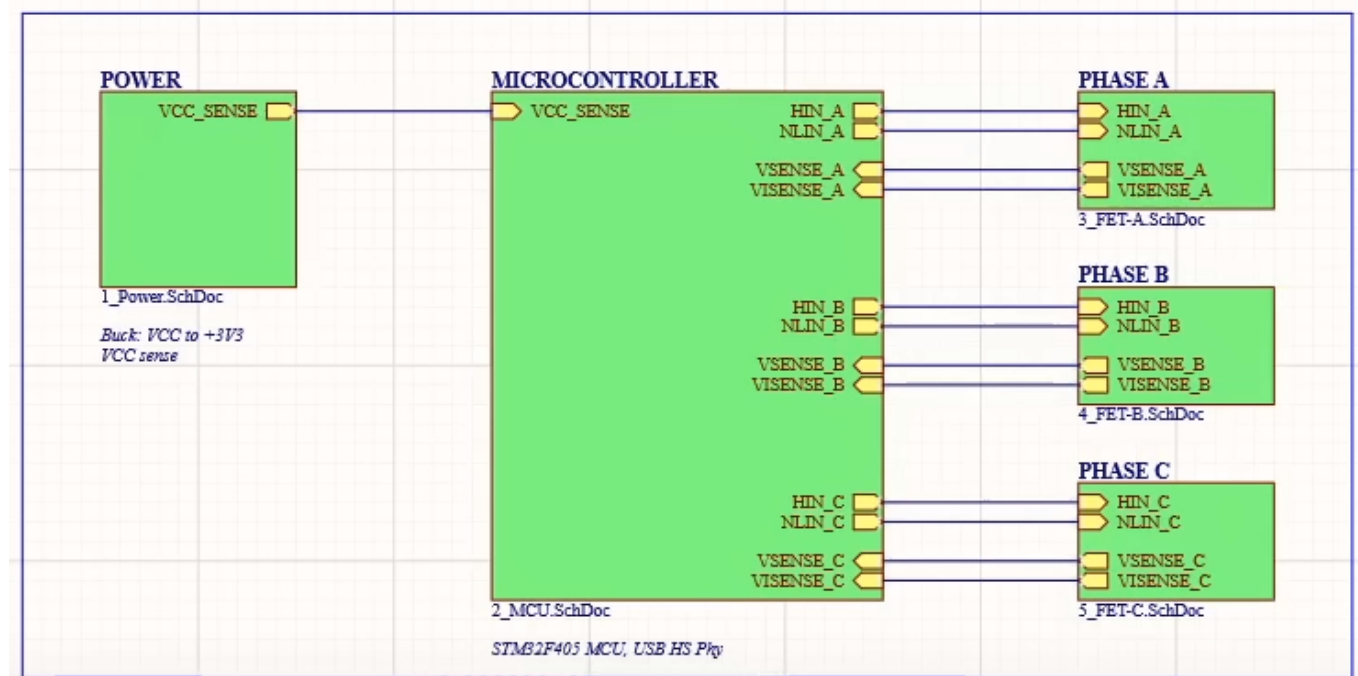
РАЗУМЕЕТСЯ, ИСПОЛЬЗОВАНИЕ ТАКИХ ГОТОВЫХ БИБЛИОТЕК И КОДОГЕНЕРАТОРОВ НЕ ДАСТ МНОГО ПОНИМАНИЯ КАК РАБОАЕТ ДВИГАТЕЛЬ И АЛГОРИТМЫ УПРАВЛЕНИЯ, НО ЗАМЕЧАТЕЛЬНО ДАСТ БУСТ ПРИ ПРОТОТИПИРОВАНИИ И ИССЛЕДОВАНИИ ПАРАМЕТРОВ ДВИГАТЕЛЯ

09.08

Для нашего конкретно использования датчики положения ротора нах** не нужны. Это несколько упрощает систему управления и

компонентную базу

Ну и общая схема драйвера упрощается примерно настолько



Первое и самое главное - необходимо отслеживать на обмотках напряжение и силу тока. Это необходимо для бездатчикового векторного управления двиглом, мониторить расход энергии и определять с какой фазы стартовать.

После двух транзисторов на фазы ставим шунтирующий резистор на входы операционника. По падению напряжения на резисторе можно определить какой ток через него течет, а операционником мы усиливаем сигнал во много раз и подаем на вход ацп МК

Делается примерно так:

