Faculty of Electrical Engineering

WARSAW UNIVERSITY OF TECHNOLOGY

www.ee.pw.edu.pl

# Od pokrętła po napęd elektryczny – enkodery inkrementalne oraz absolutne w dydaktyce

## From a knob to electric drives – incremental and absolute encoders in teaching

*XIX sympozjum „Energoelektronika w nauce i dydaktyce” – ENiD 2026*

*Centrum Nauk Technicznych UMK, Toruń, 20-22.05.2026*

Bartlomiej Ufnalski, Ph.D., D.Sc.

www.github.com/ufnalski

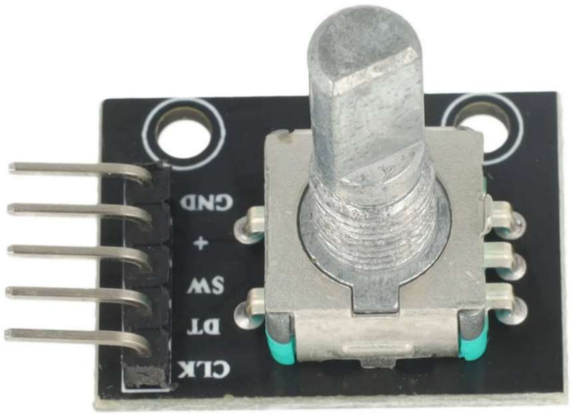
Warsaw University of Technology

Faculty of Electrical Engineering

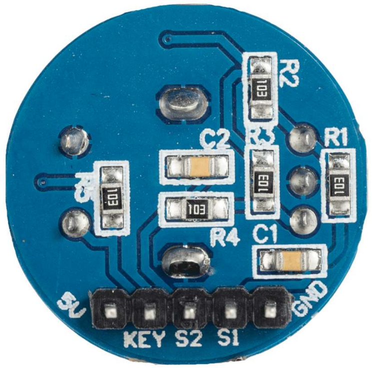
Institute of Control and Industrial Electronics

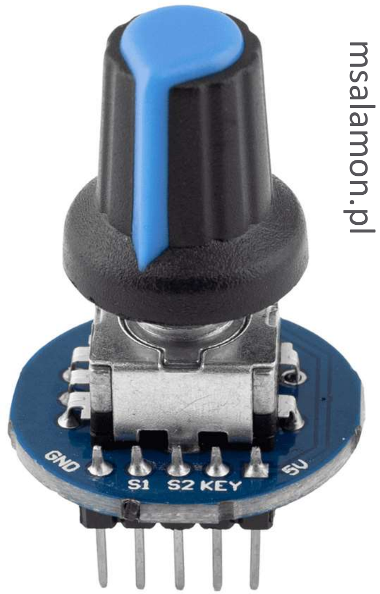
www.isep.pw.edu.pl

# Handling an encoder knob (incremental)








msalamon.pl

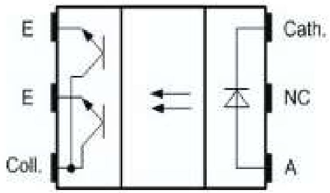
Topics:

- debouncing (software, hardware on the uC side, hardware on the encoder side)
- edge detection, direction detection (software [only to demonstrate the principle], encoder peripheral)
- edge counting (4x, 2x, 1x, 0.5x,…] and asymmetries
- speed calculation (pulse counting vs. pulse timing, delay in the feedback loop)

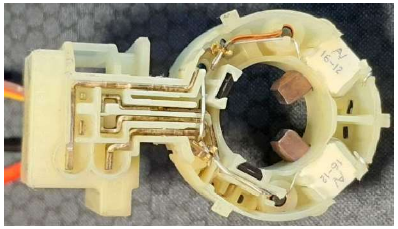
Tips:

- debouncing not only for mechanical switches: latching Hall-effect sensors

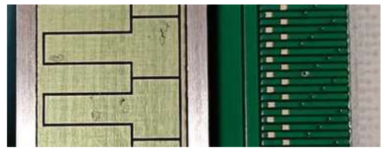
# Physical phenomena behind incremental encoders




TCUT1600X01 (Vishay)



https://github.com/ufnalski/audi_a4_tailgate_lift_motor_l432kc



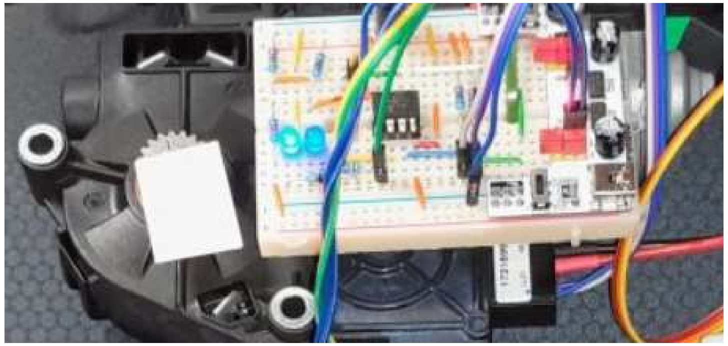
https://github.com/ufnalski/calipers_spi_g431kb (15 PLN)

Sensor types:

- mechanical
- optical
- magnetic (the Hall effect)
- capacitive
- inductive

Crankshaft position sensor (15 PLN)

# Interfaces common in incremental encoders



https://github.com/ufnalski/bmw_webasto_sunroof_drive_motor_l432kc

https://www.lika.it/eng/products/rotary-encoders/incremental/c50

ⓐ OUTPUT CIRCUITS / POWER SUPPLY

**Y2** = Push Pull, +10V÷ +30V

**L1** = Line Driver (RS422), +5V±5%

**H4** = PP/LD universal circuit, +5V÷ +30V

**L2** = Line Driver (RS422), +10V÷ +30V

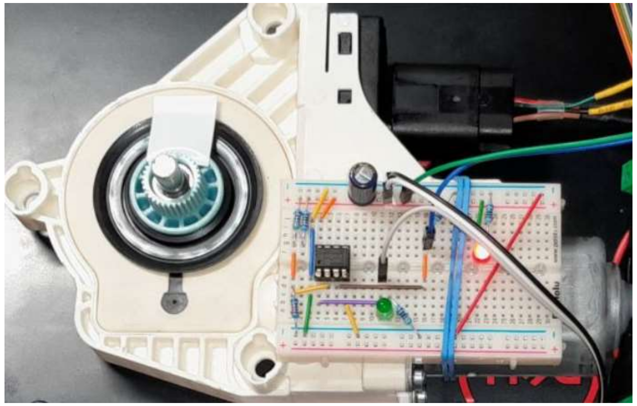

https://github.com/ufnalski/audi_a7_stabilus_powerise_l432kc

Interface types:

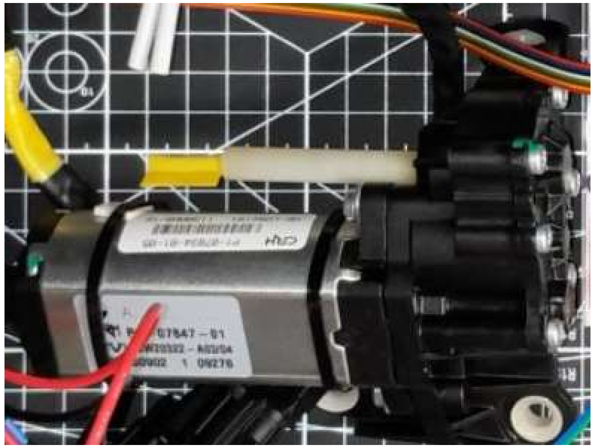
- push-pull
- open-drain (no voltage level shifters needed)
- differential (RS-422)
- current source (popular in automotive) – inspires Students to use opamps

# When drives/cars are designed by accountants

Quadrature interface? It has two channels! Do it with just one channel to make the spreadsheet happy :)



https://github.com/ufnalski/tesla_model_y_window_lift_motor_l432kc



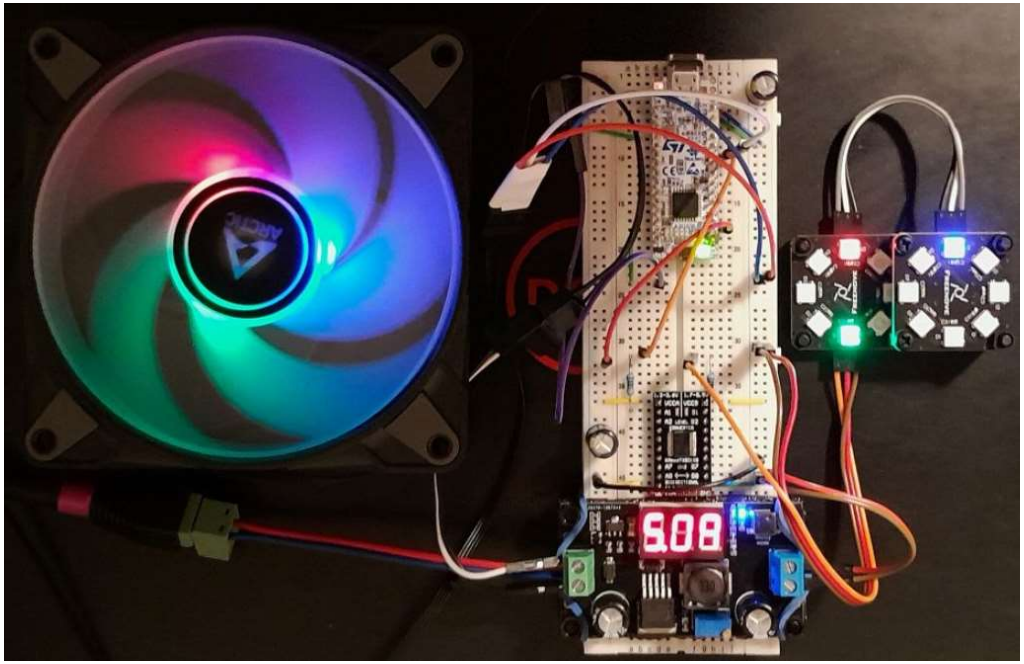
https://github.com/ufnalski/bmw_steering_column_adjust_motor_l432kc


https://github.com/ufnalski/four_pin_computer_fan_h503rb (30 PLN)


https://www.elegoo.com/blogs/arduino-projects/elegoo-tumbller-self-balancing-robot-car-tutorial

# Fun with a computer fan

Stroboscopic effect. Persistence of vision (PoV).



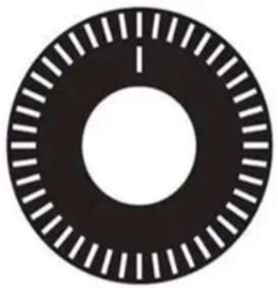

https://github.com/ufnalski/a_rgb_led_fan_strobe_g431kb
https://github.com/ufnalski/computer_fan_speed_feedback_loop_l432kc

# Absolute encoders are not (always) better

Incremental encoders facilitate:

- pulse timing to obtain speed
- interrupt triggering on count events

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim);
```



Incremental encoders can have a zero index (AB0, ABZ):

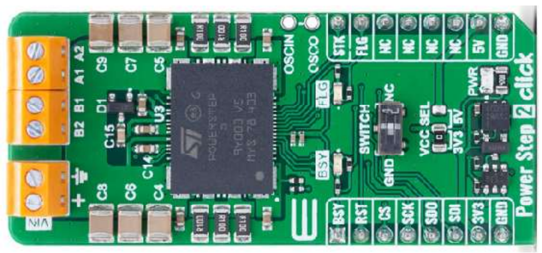
- there are applications that accept rotating the drive to finding that index before switching to the position control mode

elektroweb.pl

Are there drives that let us control position without any position sensor?

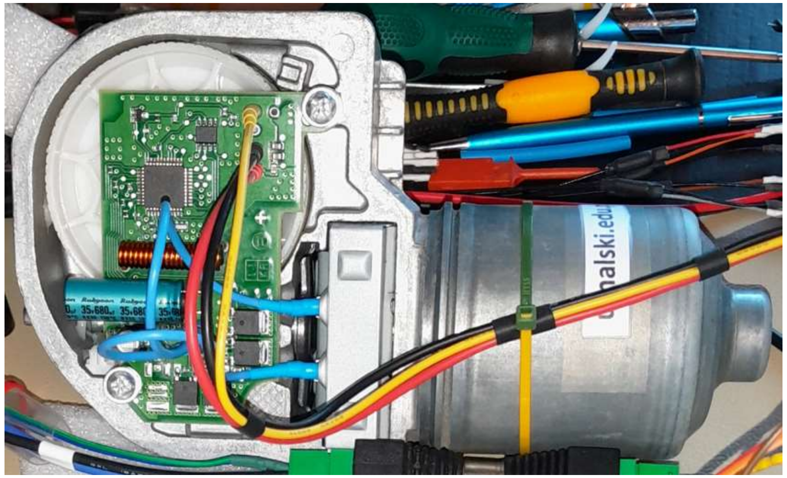
- stepper drives and microstepping – check your printers

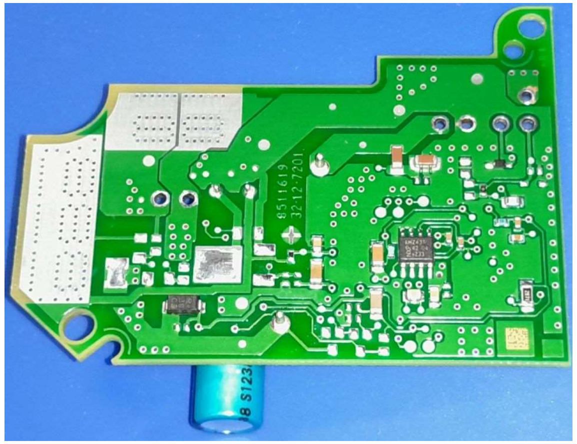
https://www.mikroe.com/power-step-2-click
https://github.com/ufnalski/stepper_motor_control_l432kc

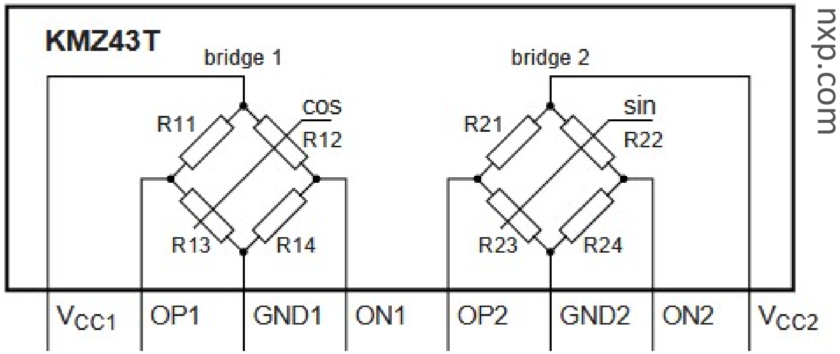


mikroe.com

# Absolute encoders



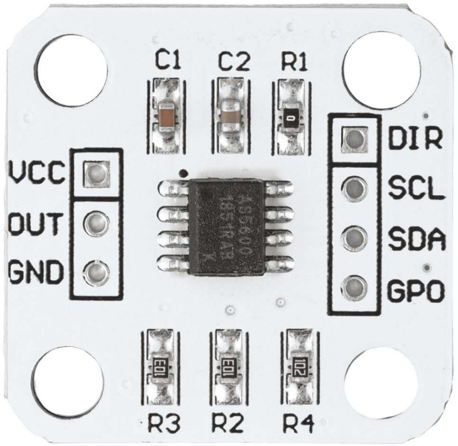



nxp.com

https://github.com/ufnalski/lin_bus_golf7_wiper_actuator_g474re
https://en.wikipedia.org/wiki/Magnetoresistance

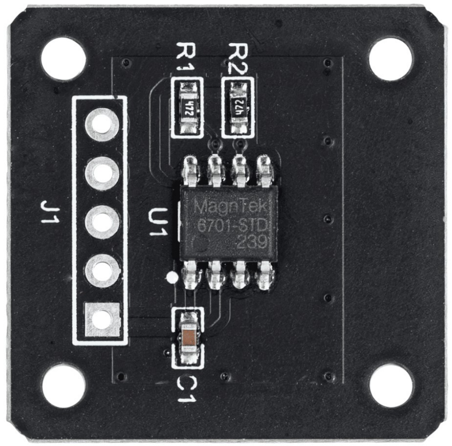
# Absolute encoders popular among DIY-ers






MagnTek
MT6701 (15 PLN)

ams OSRAM
AS5600 (10 PLN)






Interface types:
- I2C
- analog (DAC)
- PWM (TIM IC on uC side)
- SSI (Synchronous Serial Interface) – you can use SPI on uC side
- ABZ, UVW

https://github.com/ufnalski/as5600_speed_position_feedback_l432kc
https://github.com/ufnalski/mt6701_ssi_speed_feedback_l432kc

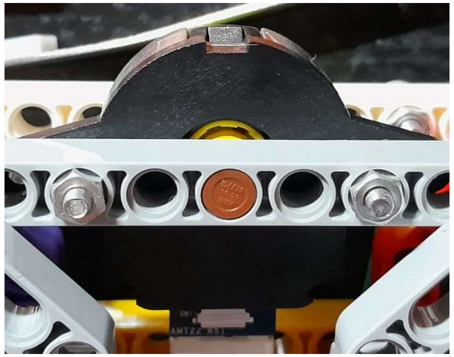
# Absolute encoders – more interfaces

More interface types:
- RS-485 (UART on uC side)
- SPI (full-duplex)
- CAN (CANopen)


sameskydevices.com

single-turn or
multi-turn versions


magnetic



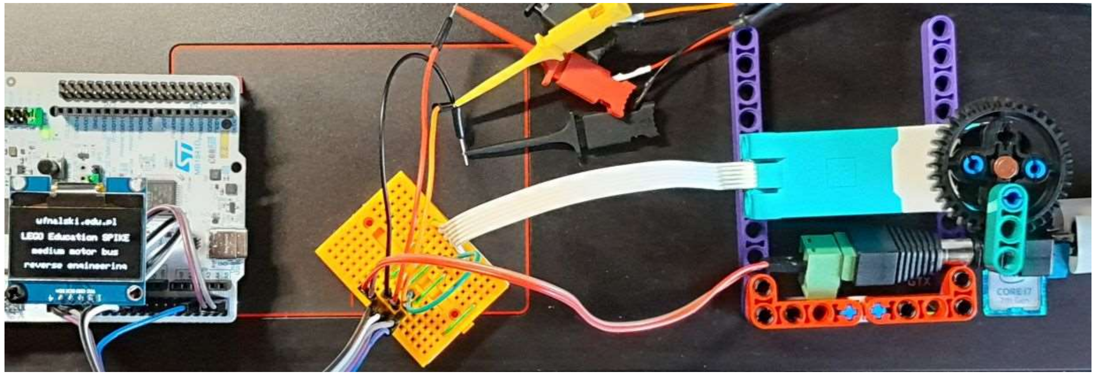
capacitive

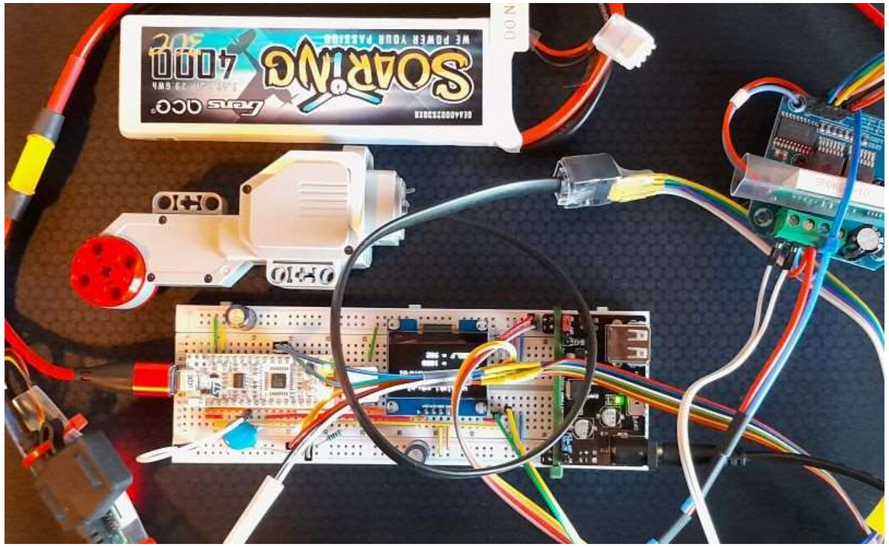
https://github.com/ufnalski/absolute_encoder_amt212dv_rs485_g431kb
https://github.com/ufnalski/absolute_encoder_amt222cv_spi_g431kb
https://github.com/ufnalski/absolute_encoder_ifm_rm9001_canopen_h533re

# Reverse engineering LEGO drives

Learning by reverse engineering:

- automotive
- home appliances
- toys

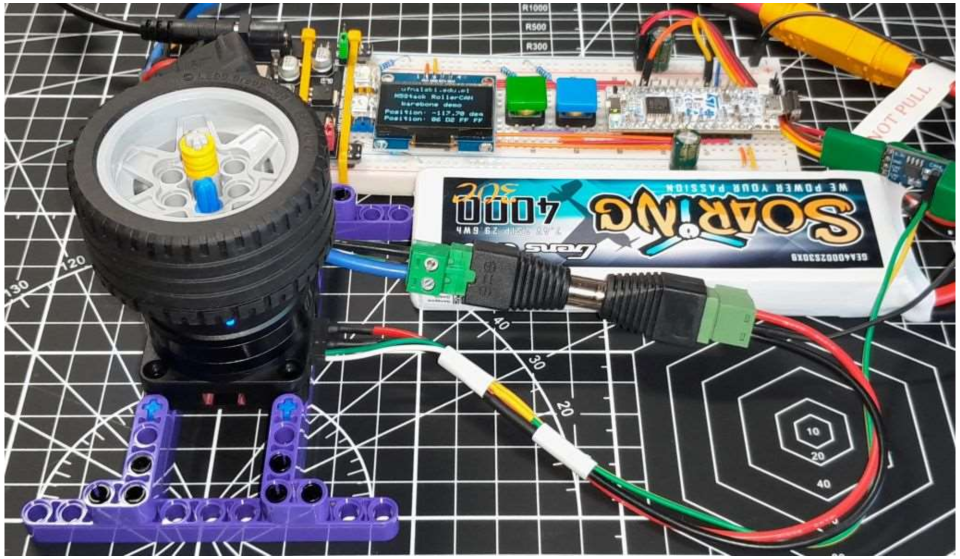


https://github.com/ufnalski/lego_ev3_large_motor_l432kc (incremental encoder)
https://github.com/ufnalski/lego_spike_medium_motor_u545re (absolute encoder)

# Open source

Learning by analyzing:
https://github.com/m5stack/M5Unit-RollerCAN-Internal-FW

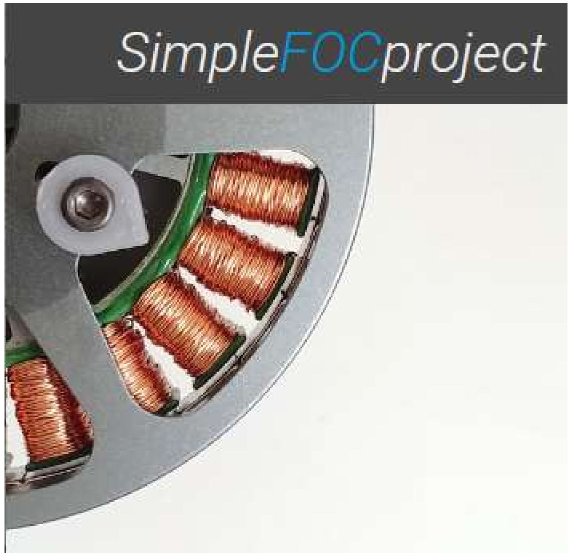



mikroe.com

https://github.com/ufnalski/m5stack_rollercan_barebone_g431kb
https://www.mikroe.com/gmr-angle-click

# Open source (cont.)

Learning by analyzing:
https://github.com/simplefoc




simplefoc.com

https://github.com/simplefoc/
Arduino-FOC-drivers

- AS5048A SPI driver - SPI driver for the AMS AS5048A absolute position magnetic rotary encoder IC.
- AS5047 SPI driver - SPI driver for the AMS AS5047P and AS5047D absolute position magnetic rotary encoder ICs.
- AS5047U SPI driver - SPI driver for the AMS AS5047U absolute position magnetic rotary encoder ICs.
- AS5600 I2C driver - I2C driver for the AMS AS5600 and AS5600L absolute position magnetic rotary encoder ICs.
- MA730 SPI driver - SPI driver for the MPS MagAlpha MA730 absolute position magnetic rotary encoder IC.
- MA730 SSI driver - SSI driver for the MPS MagAlpha MA730 absolute position magnetic rotary encoder IC.
- MA735 SPI driver - SPI driver for the MPS MagAlpha MA735 absolute position magnetic rotary encoder IC.
- MA735 SSI driver - SSI driver for the MPS MagAlpha MA735 absolute position magnetic rotary encoder IC.
- AS5145 SSI driver - SSI driver for the AMS AS5145 and AS5045 absolute position magnetic rotary encoder ICs.
- TLE5012B SPI driver - SPI (half duplex) driver for TLE5012B absolute position magnetic rotary encoder IC.
- STM32 Hardware Encoder - STM32 Hardware timer based encoder driver for ABI type quadrature encoders.
- ESP32 Hardware Encoder - ESP32 Hardware timer based encoder driver for ABI type quadrature encoders.
- SC60228 SPI driver - SPI driver for SemiMent SC60288 magnetic encoder IC.
- MA330 SPI driver - SPI driver for the MPS MagAlpha MA330 absolute position magnetic rotary encoder IC.
- MT6816 SPI driver - SPI driver for the MagnTek MT6816 absolute position magnetic rotary encoder IC.
- MT6701 SSI driver - SSI driver for the MagnTek MT6701 absolute position magnetic rotary encoder IC.
- MT6835 SPI driver - SPI driver for the MagnTek MT6835 21 bit magnetic rotary encoder IC.
- STM32 PWM sensor driver - STM32 native timer-based driver for PWM angle sensors.
- CalibratedSensor - A sensor which can calibrate for eccentricity on the magnet placement.
- SmoothingSensor - A SimpleFOC Sensor wrapper implementation which adds angle extrapolation.

# Absolute encoder – speed calculation

$$\omega(t) = \frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} \cong \frac{\Delta\alpha}{\Delta t}$$

The control system is never fed by the actual speed – a delay is no shorter than $\frac{1}{2}\Delta t$.
The delay **has** to be taken into account when tuning the speed control loop.

- For most absolute encoders we usually go for $\Delta t = \mathrm{const}$ (polling, no EXTI).
- Averaging acts as a low-pass filter. An additional, more elaborate low-pass filter can be added. Still, it introduces more delay (phase shift) and amplitude attenuation. The trade-off is alway there.

Why such an electric motor? The magic behind iconic brands to engage our Students :)

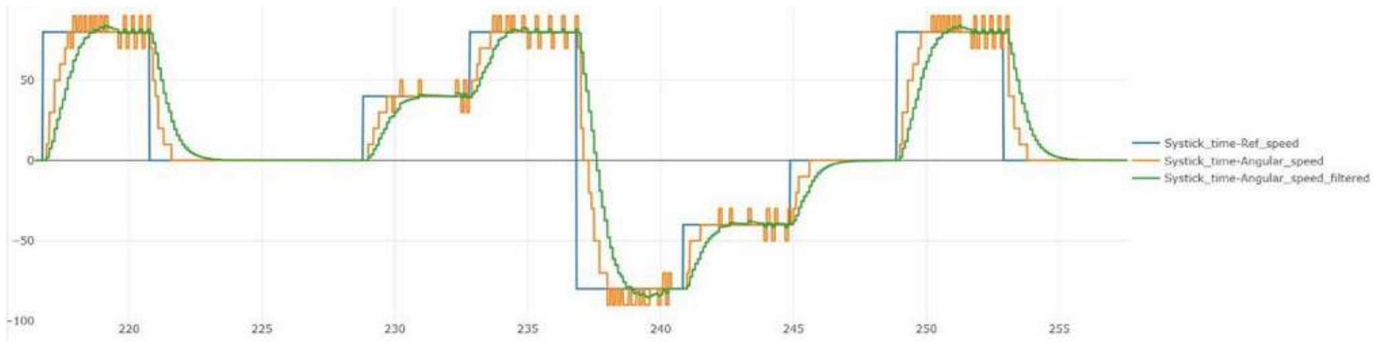
https://github.com/ufnalski/as5600_speed_position_feedback_l432kc

# Incremental encoder – speed calculation (pulse counting)

$$\omega(t) = \frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} \cong \frac{\Delta\alpha}{\Delta t} = \frac{2\pi n}{N_{\mathrm{CPR}}\Delta t} = \omega(n)$$

- I quite often come across implementations that use pulse counting to calculate speed. Why poll the quadrature peripheral when the count events can trigger a timer responsible for pulse timing? Is it kind of borrowing from absolute encoders?
- Then I put it into some perspective for my Students. For example, the STM32C562RE can run at 144 MHz. Let's consider a drive with the max speed of 6000 rpm (just to make my calculations easier), i.e. 100 rps. Now let's assume some demanding scenario regarding pulse timing, e.g. 10000 counts per revolution. This would give 1 mln counts per second. The relative resolution of time measurement is still under 0.7%.
- And watch what happens at low speeds and/or for low-CPR encoders if the time window is not adaptable (adjustable).

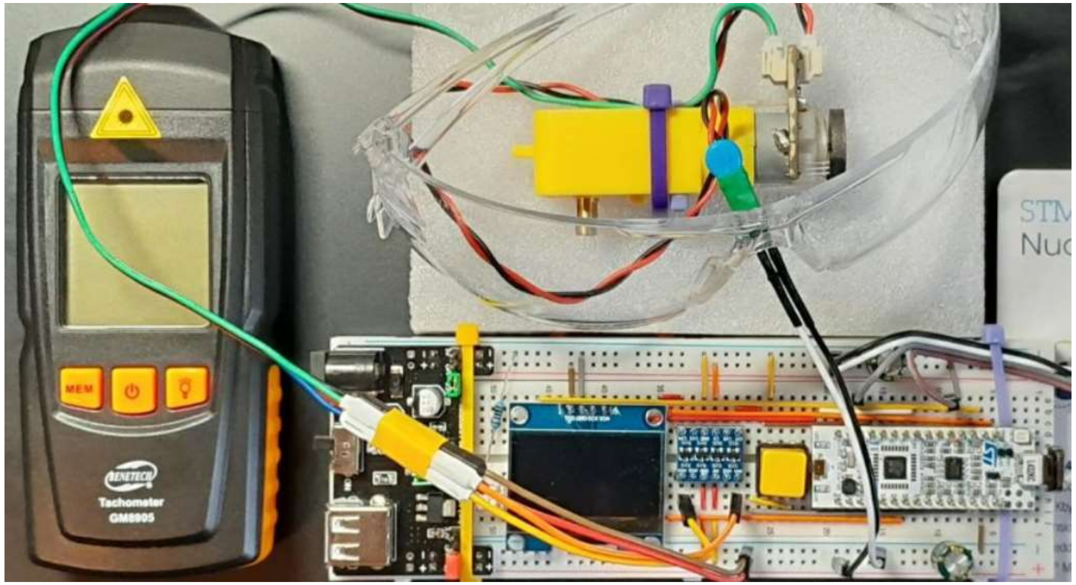
https://github.com/ufnalski/bmw_steering_column_adjust_motor_l432kc



# Incremental encoder – speed calculation (pulse timing)

$$\omega(t) = \frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} \cong \frac{\Delta\alpha}{\Delta t} = \frac{2\pi n}{N_{\mathrm{CPR}}\Delta t_n} = \frac{2\pi}{N_{\mathrm{CPR}}\Delta t_1}$$

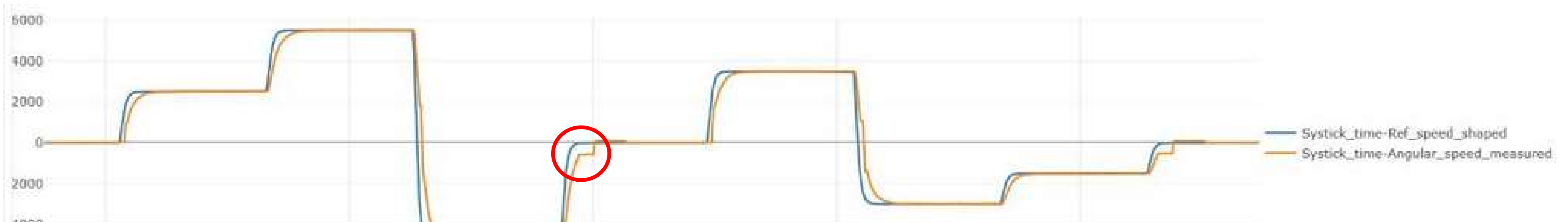
- In my humble opinion, pulse timing should be regarded as a default approach (if no switching between the two methods is envisaged).
- Pulse timing can (and should) be done entirely in the peripheral (TIM/QEP), i.e. without interrupting the main core.
- The length of the counter, e.g. of 16 bits, is not a limitation for (very) low speeds – you can always count overflow events and put them into the formula. You can easily use an interrupt-triggered software counter to spare the hardware one: e.g. 144 MHz / 2^16 < 2 kHz.



https://github.com/ufnalski/incremental_encoder_pulse_timing_l432kc
https://www.ti.com/lit/ug/sprug74/sprug74.pdf

# Incremental encoder – speed calculation (pulse timing) and fixed speed control loop sampling

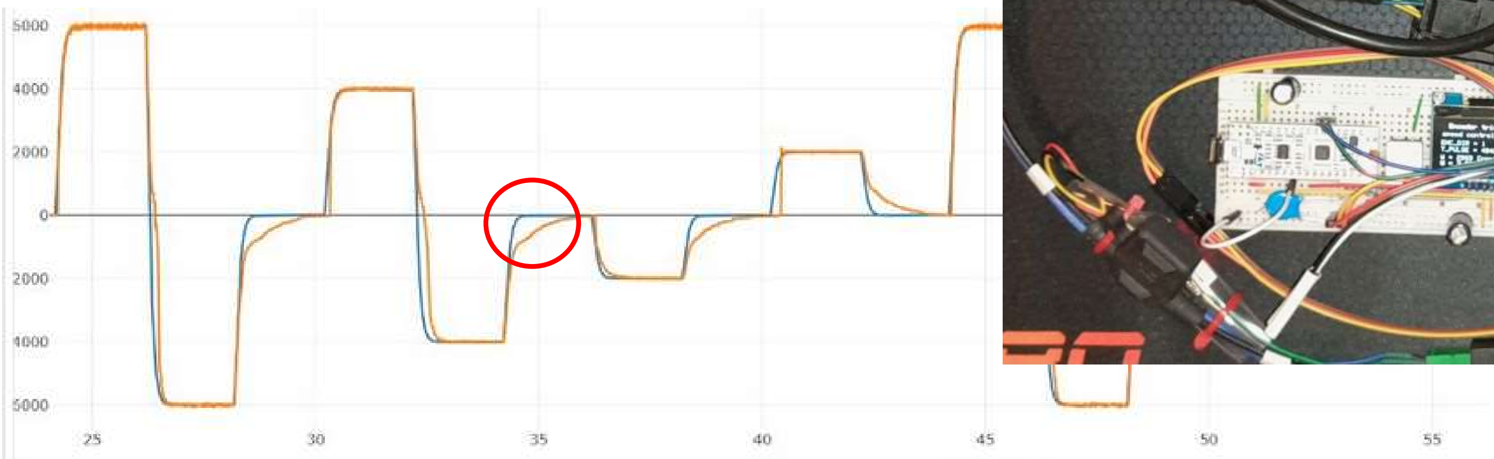
- regarded as a classical approach
- speed calculation code needs special handling if no encoder pulses are present (thus no interrupt callbacks are executed)

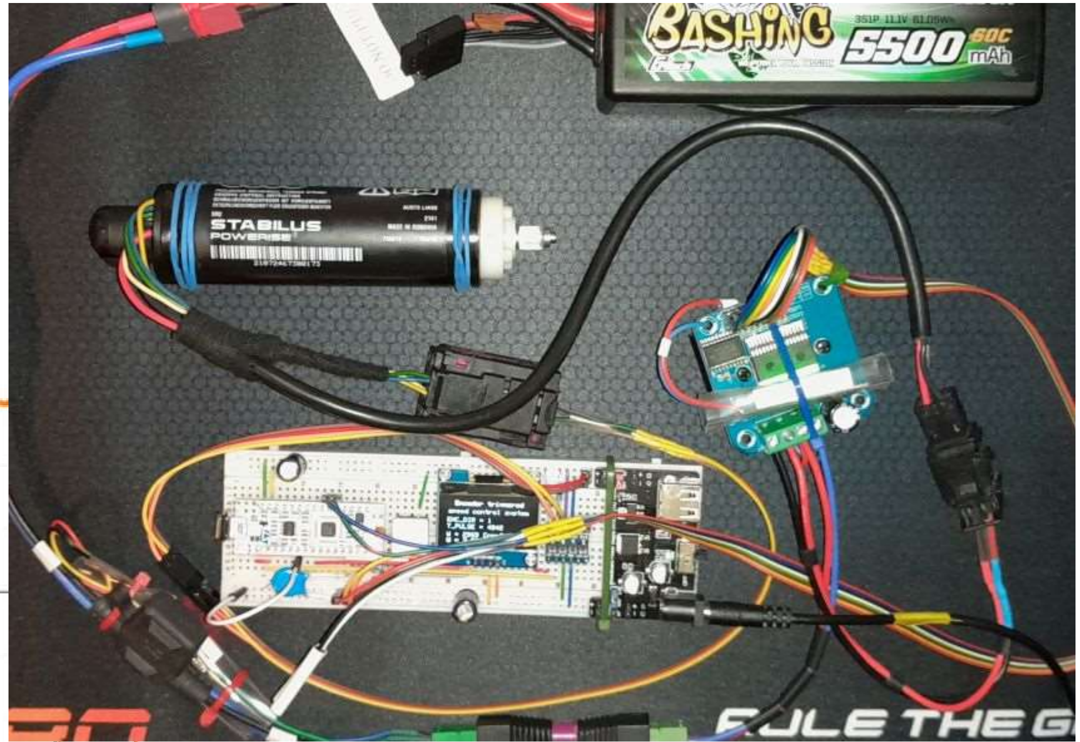




https://github.com/ufnalski/audi_a4_tailgate_lift_motor_l432kc

# Incremental encoder – speed calculation (pulse timing) and encoder-triggered speed control loop execution

- smallest possible delay (if no model-based algorithms are deployed)
- variable delay -> variable parameters
- low-CPR encoders (prevalent in many actuators for the body control module (automotive industry) are a perfect choice to play with the concept



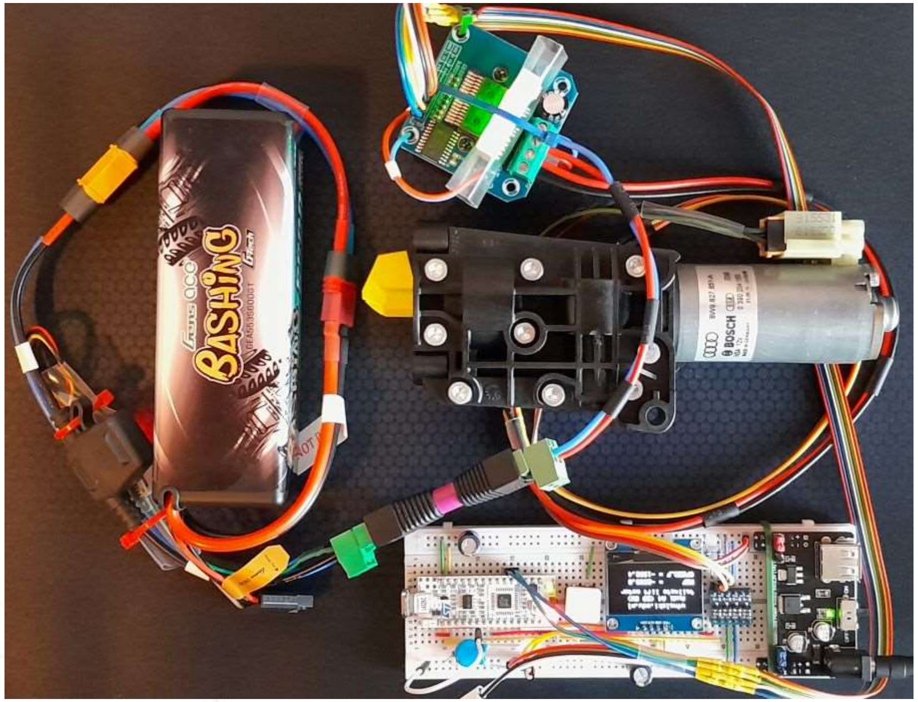

https://github.com/ufnalski/encoder_triggered_speed_feedback_l432kc

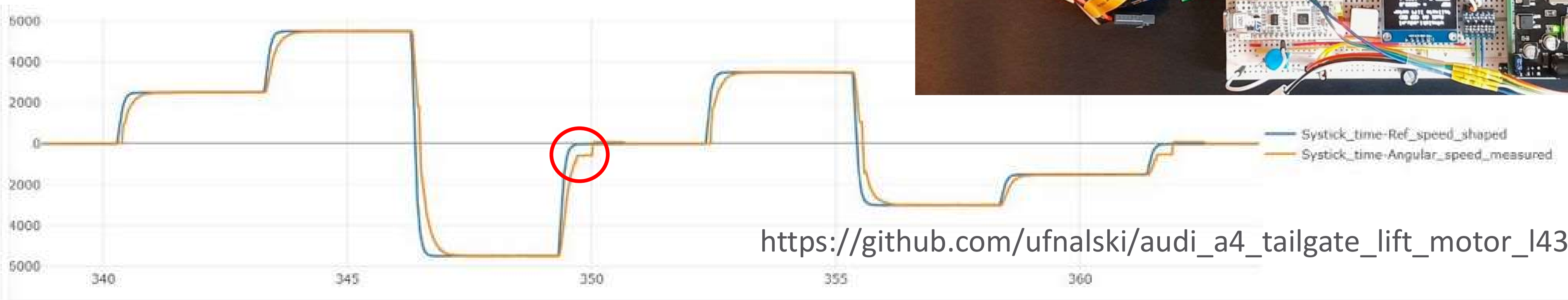
# Learning by discovery

The code available in the repository usually:

- provides some initial steps to facilitate moving faster and not breaking things, at least not all of them, along the way,
- does not contain a complete solution, rather illustrates the problem at hand.

This is organised in such a way on purpose – the goal is to leave some unwanted features/characteristics (resulting in poor performance) to be discovered by my trainees, and then to be corrected during the workshop meetings.





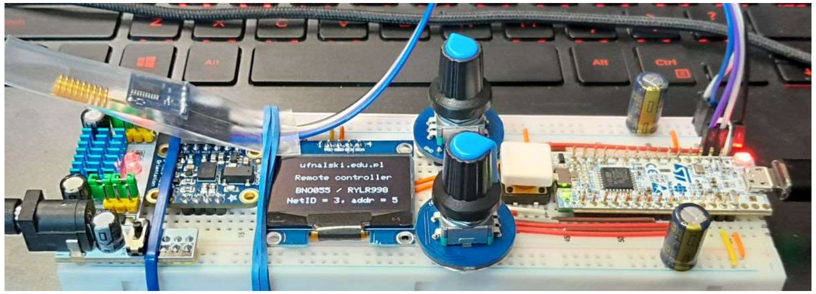

https://github.com/ufnalski/audi_a4_tailgate_lift_motor_l432kc

# Overwhelmed by the variety of encoders?

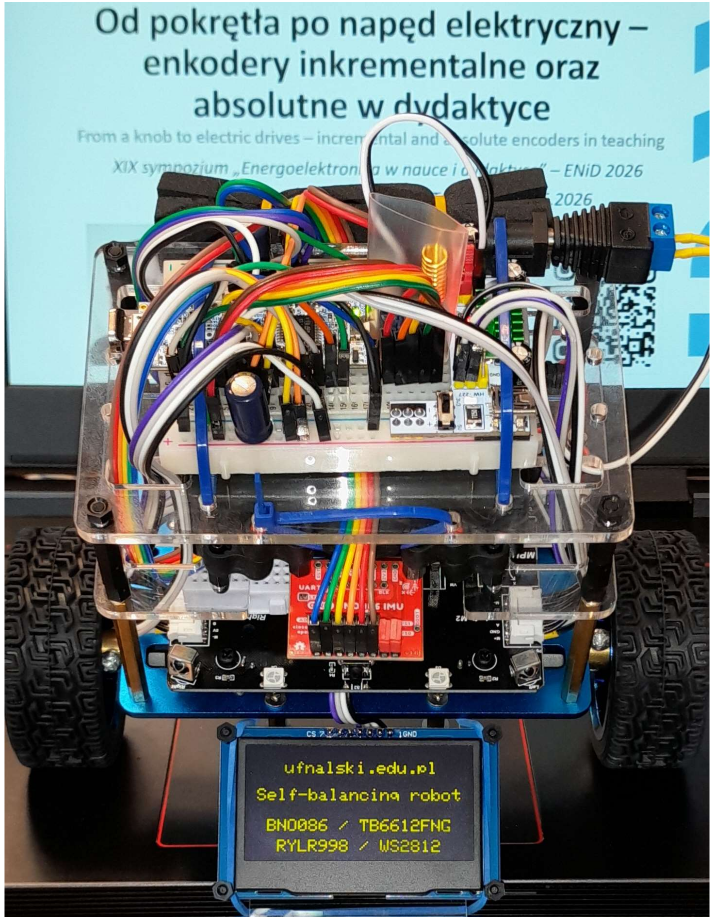
Don’t worry. Just start playing with them, and the skill to choose the right one for your application will follow. The beauty of hobby projects is that they usually offer a high level of flexibility in choosing components – every dollar spent will not be then multiplied by 10 000 000 for a mass-produced drive such as a wiper actuator.

**From a basic encoder knob, through a fancy haptic rotary knob, to a self-balancing robot, or any other motion control system of your choice – enjoy the journey into position sensors for electric drives!**

m5stack.com



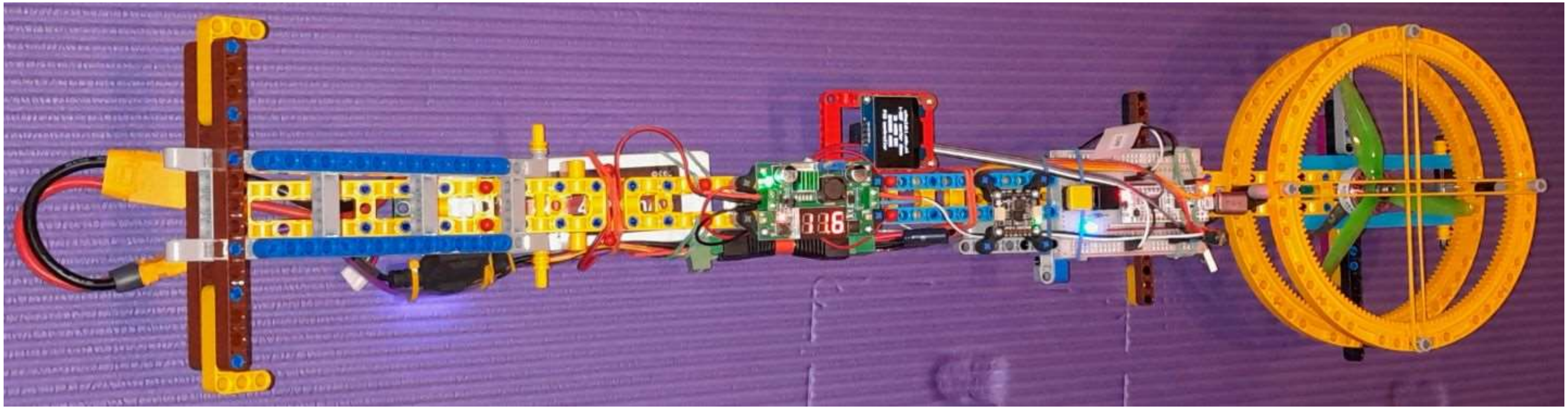
https://github.com/ufnalski/remote_controller_rylr998_bno055_g431kb
https://github.com/ufnalski/m5stack_rollercan_barebone_g431kb




arachnohobbia

# Go sensorless (if low speeds are not needed)

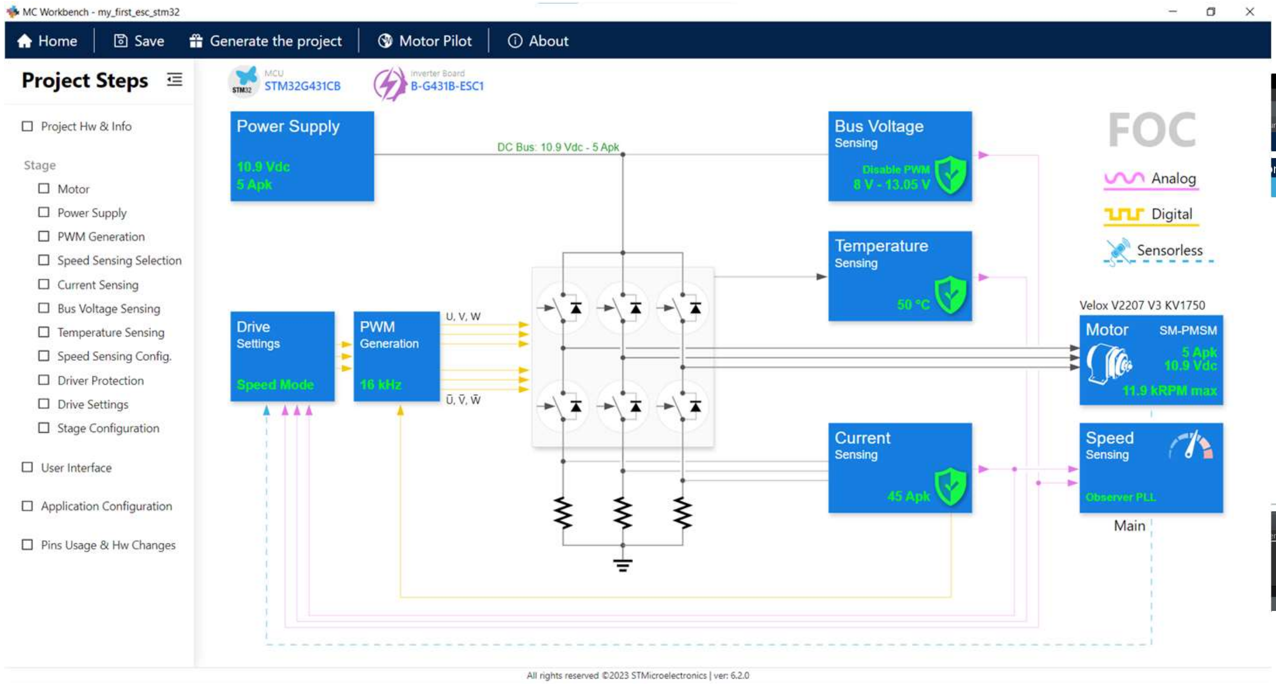
BLDC ESC (electronic speed controller) for drones



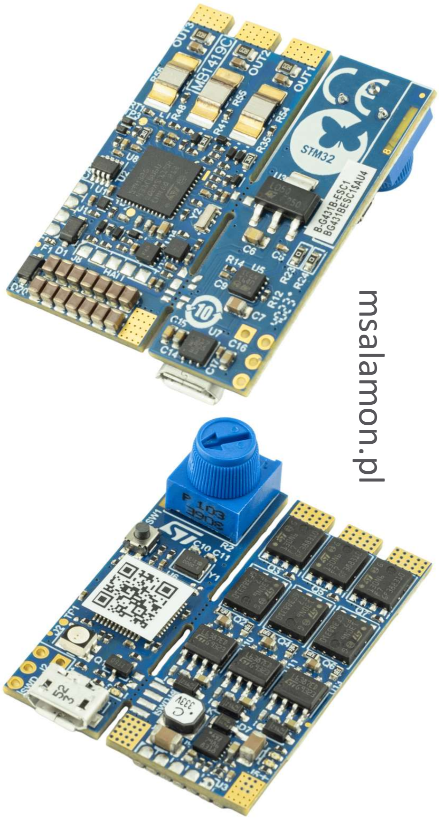
https://github.com/ufnalski/one_dof_copter_bno085_l432kc
https://github.com/am32-firmware/AM32

# Open source (cont.)

Learning by analyzing.








https://www.st.com/en/evaluation-tools/b-g431b-esc1.html
https://wiki.st.com/stm32mcu/wiki/STM32MotorControl:STM32_MC_Workbench
http://ufnalski.edu.pl/stm32_projects/motor_control_workbench/
https://github.com/pat92fr/BrushlessServoController

115 PLN

# Encoder as a great teaching aid not only in the field of electric drives

One can play in an engaging way with many MCU peripherals:

- I2C
- SPI/SSI
- UART
- CAN
- CRC and other checksums
- TIMER Encoder Mode
- TIMER Input Capture
- TIMER Output Compare (to catch overflows)
- NVIC (Nested Vectored Interrupt Controller), EXTI

```
while (_n_of_bits--)
{
    _data = _data & 0x80000000 ? (_data << 1) ^ 0x0c000000 : _data << 1;
}
return _data >> 26;
```

```
uint8_t csum0 = 0;
uint8_t csum1 = 0;
for (uint8_t i = 0; i < 8; i++)
{
    csum0 ^= (data >> i * 2) & 1;
    csum1 ^= (data >> (i * 2 + 1)) & 1;
}
return (csum0 && csum1);
```

https://github.com/ufnalski/mt6701_ssi_speed_feedback_l432kc
https://github.com/ufnalski/absolute_encoder_amt232av_ssi_g431kb


# Thank you for your kind attention!

**WARNING**

**Incremental and absolute encoders – do try them at home!**

# Presentation available in PDF

http://ufnalski.edu.pl/proceedings/enid2026/incremental_and_absolute_encoders.pdf

# Open Day at the Faculty of Electrical Engineering

http://ufnalski.edu.pl/control_engineering_for_hob
byists/2026_dzien_otwarty_we/Dzien_Otwarty_WE
_2026_Control_Engineering_for_Hobbyists.pdf
**https://www.ee.pw.edu.pl/dzienotwarty/**
**https://app.evenea.pl/event/911264-1/**

# AI is not going to replace power electronics engineers (soon)

**Studia techniczne dają stabilną przyszłość. Inżynierowie i energetycy nie narzekają na brak ofert**

Rozwój technologii, automatyzacji i odnawialnych źródeł energii powoduje wzrost zapotrzebowania na specjalistów technicznych. Firmy szukają osób zajmujących się energetyką, automatyką czy instalacjami elektrycznymi. Branże techniczne należą dziś do najszybciej rozwijających się sektorów gospodarki.

Dobrym wyborem mogą być studia z elektrotechniki, energetyki albo automatyki i robotyki.




https://next.gazeta.pl/next/7,151003,32795865,po-tych-studiach-szybko-znajdziesz-prace-te-zawody-beda-rozchwytywane.html
https://perspektywy.pl/poradnik-maturzysty/czego-warto-sie-uczyc-poznaj-top-10-zawodow-w-2026-roku
https://www.pulshr.pl/przemysl/dla-nich-praca-bedzie-na-pewno-te-zawody-zdominuja-rynek-pracy,115380.html