

# Brushless PM Motor Control using Cycle Accurate ARM Model in COSIDE®

Hayri Hasou, Infineon Technologies Italy  
November 2021



# Table of contents

---

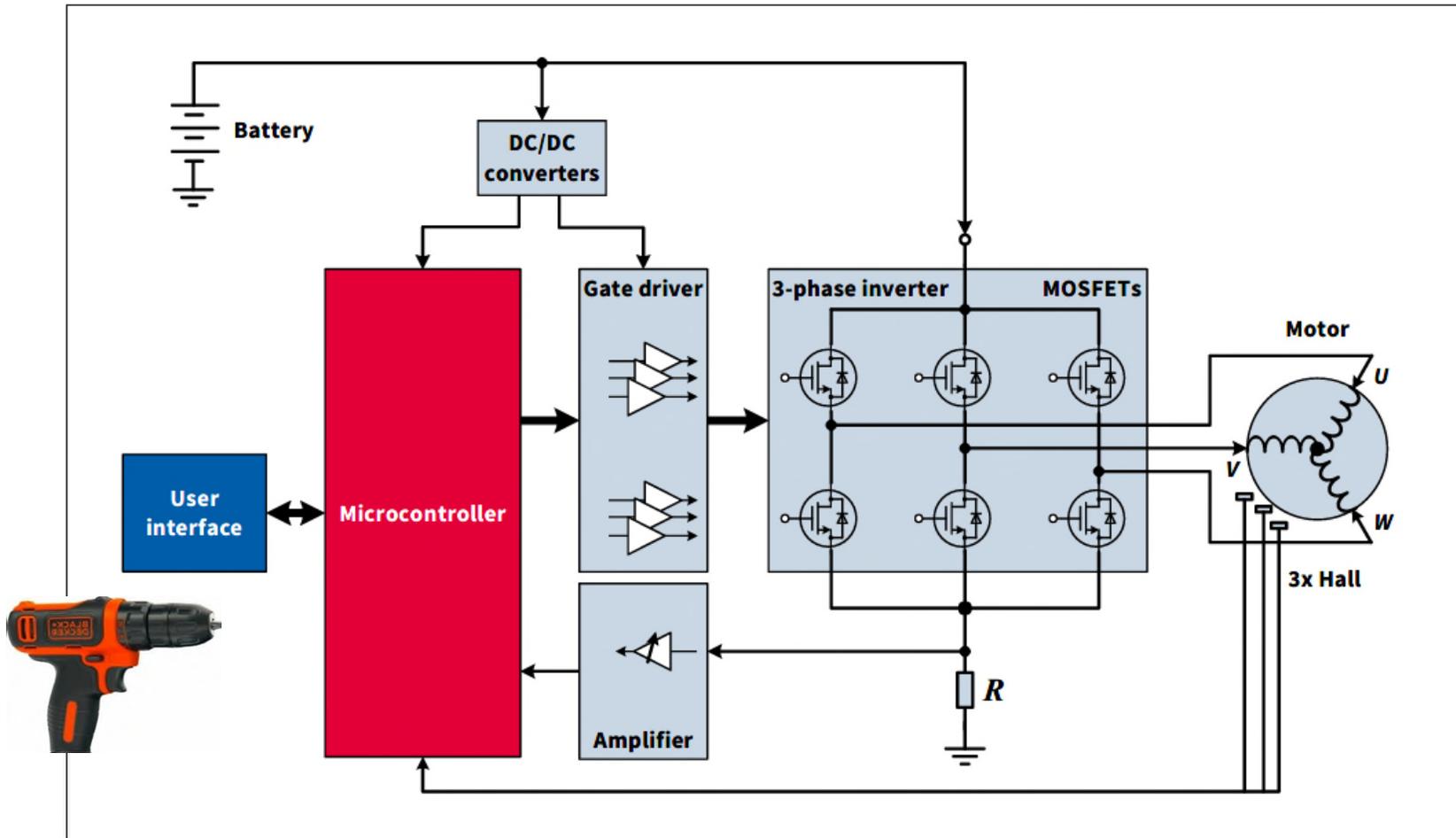
|   |                                   |    |
|---|-----------------------------------|----|
| 1 | System introduction               | 3  |
| 2 | Trapezoidal motor characteristics | 6  |
| 3 | Block commutation                 | 10 |
| 4 | ARM Cortex M0+ integration        | 15 |

# Table of contents

---

|   |                                   |    |
|---|-----------------------------------|----|
| 1 | System introduction               | 3  |
| 2 | Trapezoidal motor characteristics | 6  |
| 3 | Block commutation                 | 10 |
| 4 | ARM Cortex M0+ integration        | 15 |

# A system example: Power Tool



## Use case:

Screwing-In and Out a screw in a dowel





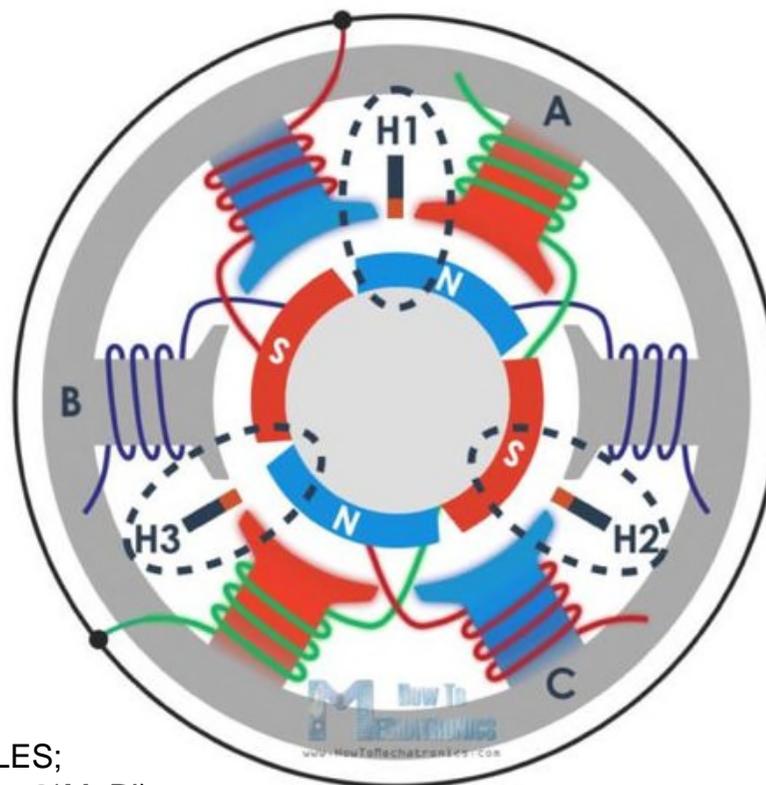
# Table of contents

---

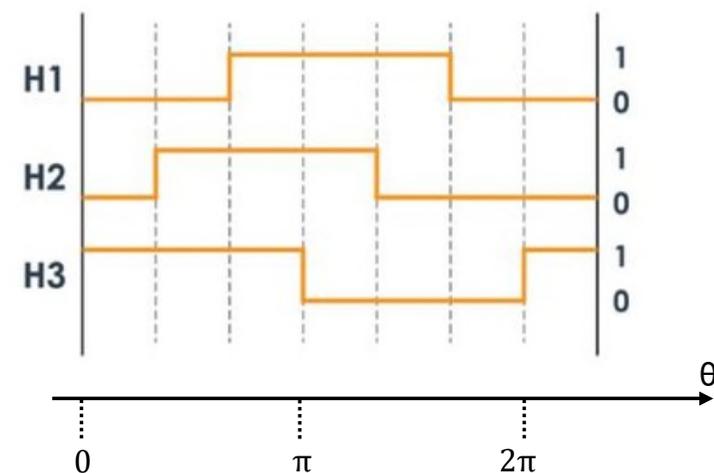
|   |                                   |    |
|---|-----------------------------------|----|
| 1 | System introduction               | 3  |
| 2 | Trapezoidal motor characteristics | 6  |
| 3 | Block commutation                 | 10 |
| 4 | ARM Cortex M0+ integration        | 15 |

# BLDC motor with Hall sensors

4 POLES motor



Hall-effect Sensors Output



```

////////////////////////////////////
// SC_METHOD rot_angle
// sensitive << ang_p;
////////////////////////////////////
void load_control::rot_angle()
{
    double ang_e;
    double total_ang_e;
    total_ang_e = ang_p / 2 * POLES;
    ang_e = std::fmod(total_ang_e, 2 * M_PI);
    hall_a = (M_PI * 5 / 3 <= ang_e || ang_e < M_PI * 2 / 3) ? true : false;
    hall_b = (M_PI * 1 / 3 <= ang_e && ang_e < M_PI * 4 / 3) ? true : false;
    hall_c = (M_PI * 3 / 3 <= ang_e && ang_e < M_PI * 6 / 3) ? true : false;
}
    
```

# Motor Equations

For each  $V_a, V_b, V_c$

$$V_n = R_i \cdot I_n + L \cdot \frac{dI_n}{dt} + E_n$$

$$\theta_e = P/2 \cdot \theta_m$$

$$E_n = k_e \cdot \omega_m \cdot F_n(\theta_e)$$

$$T_n = k_t \cdot I_n \cdot F_n(\theta_e)$$

## Parameters

| Name                    | Type   | Default | Description                   |
|-------------------------|--------|---------|-------------------------------|
| $L_{\text{phase}}$      | double | 0.28e-3 | inductance in Henry           |
| $r_{i\text{phase}}$     | double | 0.6     | resistance in Ohm             |
| $k_e$                   | double | 25.5e-3 | back-emf constant in (Vs)/rad |
| $k_t$                   | double | 25.5e-3 | torque constant in Nm/A       |
| $\text{initial\_angle}$ | double | 0.0     | initial angle in degree       |

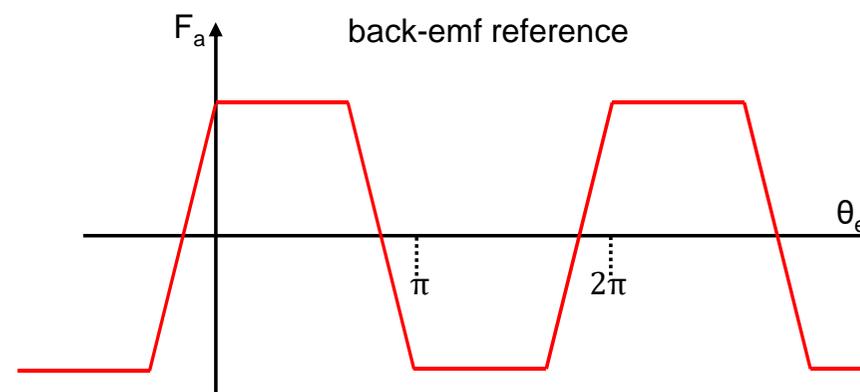
## Back-emf reference

$$F_a(\theta_e) \quad \theta_e = 0 \dots \frac{2}{3}\pi \quad \rightarrow 1$$

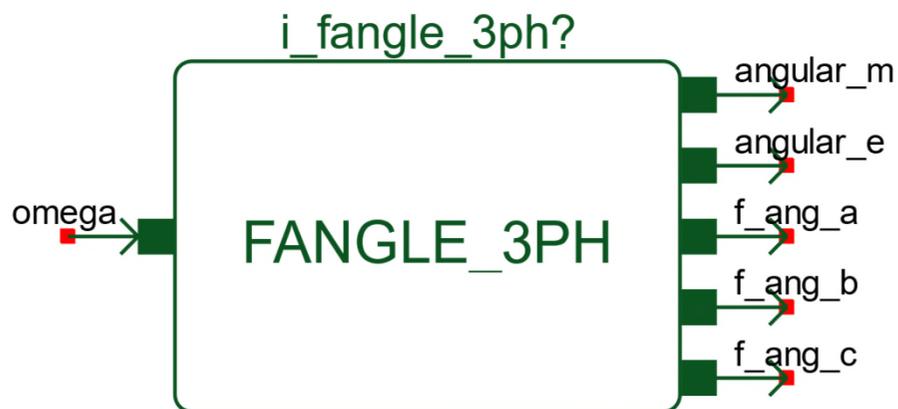
$$\theta_e = \frac{2}{3}\pi \dots \pi \quad \rightarrow 1 - \frac{6}{\pi} \left( \theta_e - \frac{2}{3}\pi \right)$$

$$\theta_e = \pi \dots \frac{5}{3}\pi \quad \rightarrow -1$$

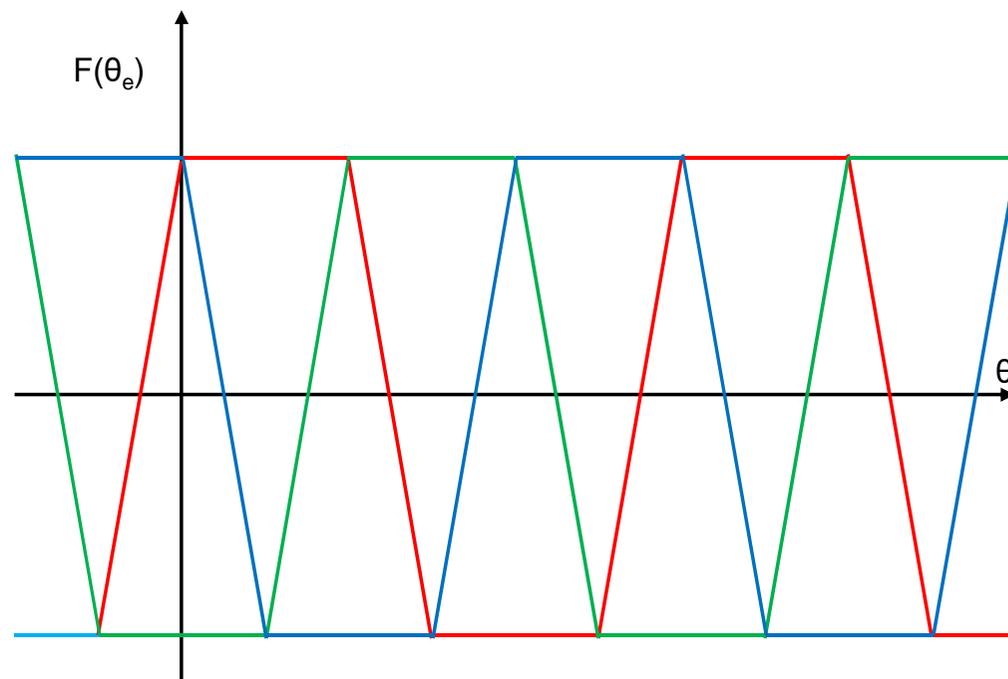
$$\theta_e = \frac{5}{3}\pi \dots 2\pi \quad \rightarrow -1 + \frac{6}{\pi} \left( \theta_e - \frac{5}{3}\pi \right)$$



# BEMF reference (sca\_mechanical\_libraries->motors->fangle\_3ph)



The module calculates the back-emf reference for a 2 Poles 3-phase trapezoidal BLDC motor. Initial angle is a module parameter



## Ports

| Name      | Interface        | Type   | Description        |
|-----------|------------------|--------|--------------------|
| omega     | sca_tdf::sca_in  | double | angular speed      |
| angular_m | sca_tdf::sca_out | double | current angular    |
| angular_e | sca_tdf::sca_out | double | current angular    |
| f_ang_a   | sca_tdf::sca_out | double | f(angular) phase a |
| f_ang_b   | sca_tdf::sca_out | double | f(angular) phase a |
| f_ang_c   | sca_tdf::sca_out | double | f(angular) phase a |

# Table of contents

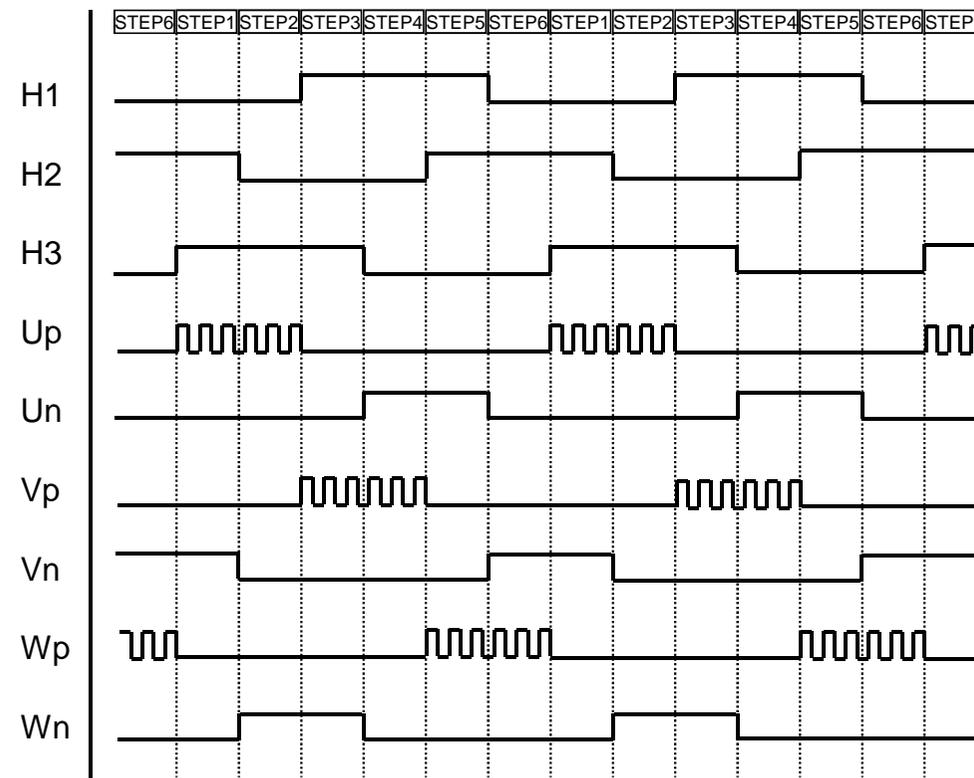
---

|   |                                   |    |
|---|-----------------------------------|----|
| 1 | System introduction               | 3  |
| 2 | Trapezoidal motor characteristics | 6  |
| 3 | Block commutation                 | 10 |
| 4 | ARM Cortex M0+ integration        | 15 |

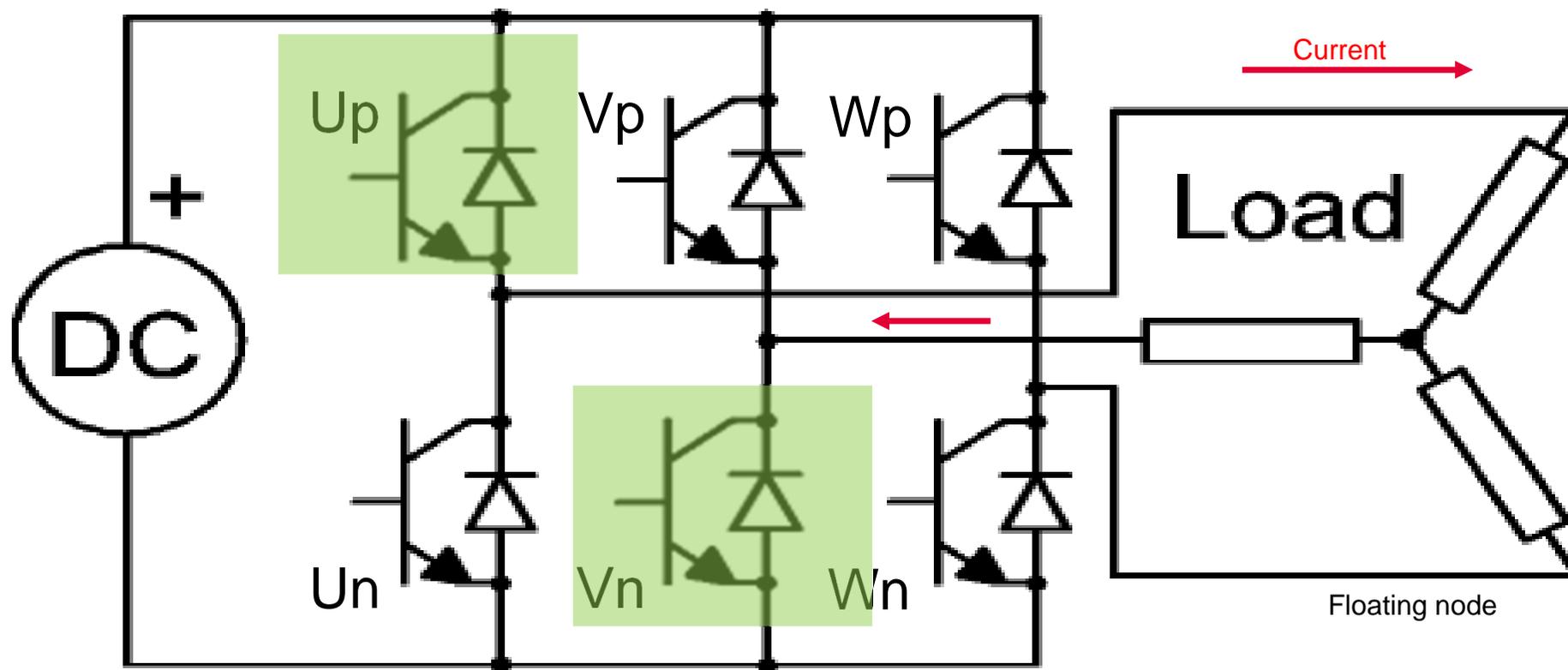
# Six steps commutation

Active switch

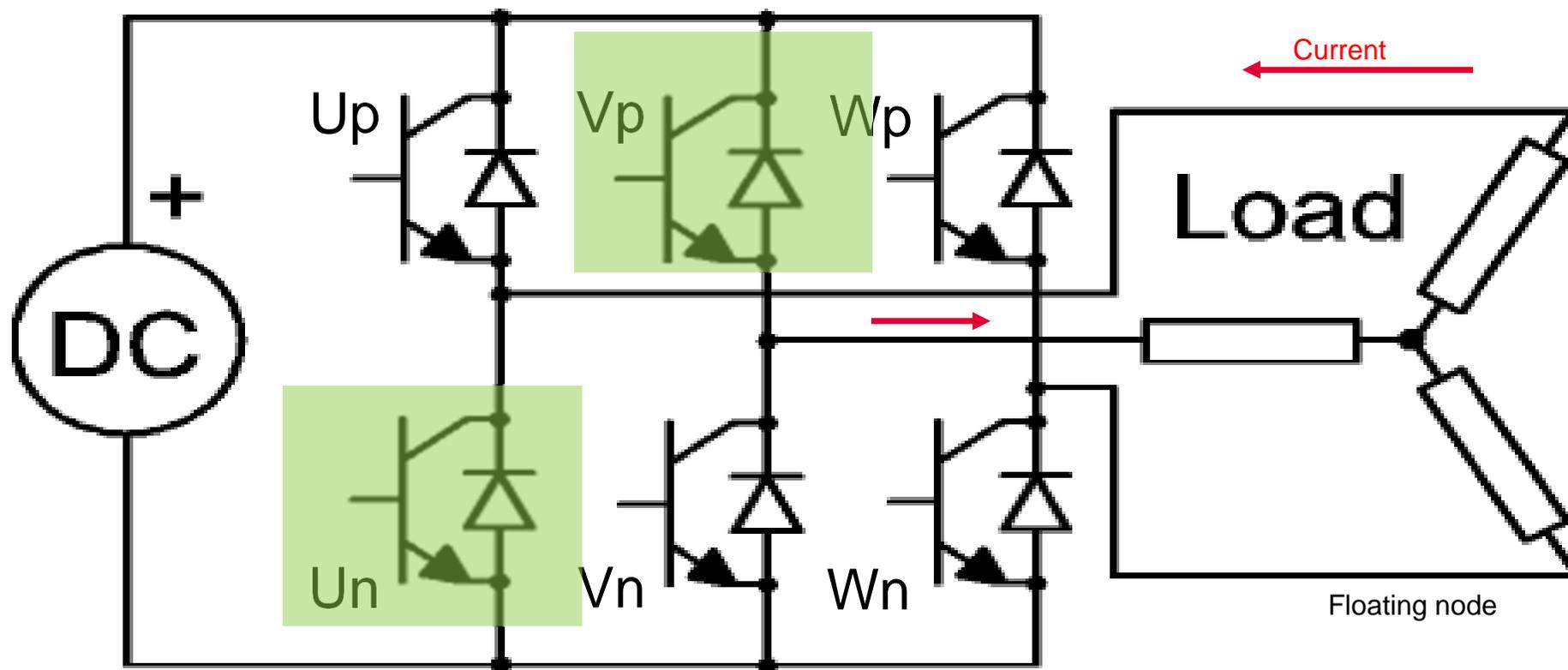
|       |    |    |    |  |
|-------|----|----|----|--|
| STEP1 | Up | Vp | Wp |  |
|       | Un | Vn | Wn |  |
| STEP2 | Up | Vp | Wp |  |
|       | Un | Vn | Wn |  |
| STEP3 | Up | Vp | Wp |  |
|       | Un | Vn | Wn |  |
| STEP4 | Up | Vp | Wp |  |
|       | Un | Vn | Wn |  |
| STEP5 | Up | Vp | Wp |  |
|       | Un | Vn | Wn |  |
| STEP6 | Up | Vp | Wp |  |
|       | Un | Vn | Wn |  |



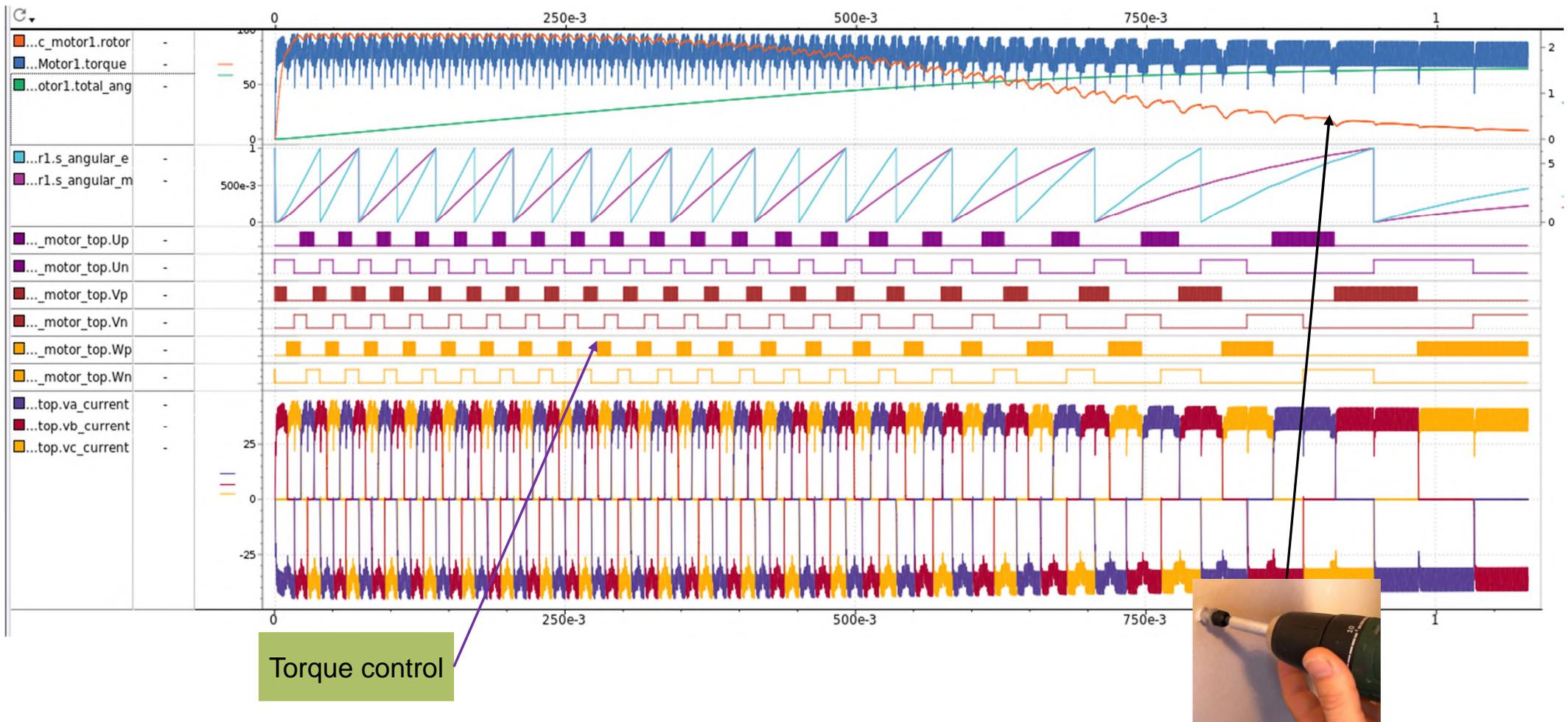
# Step 1 (UV)



# Step 4 (VU)



# Motor starting and slowing down as load is increased

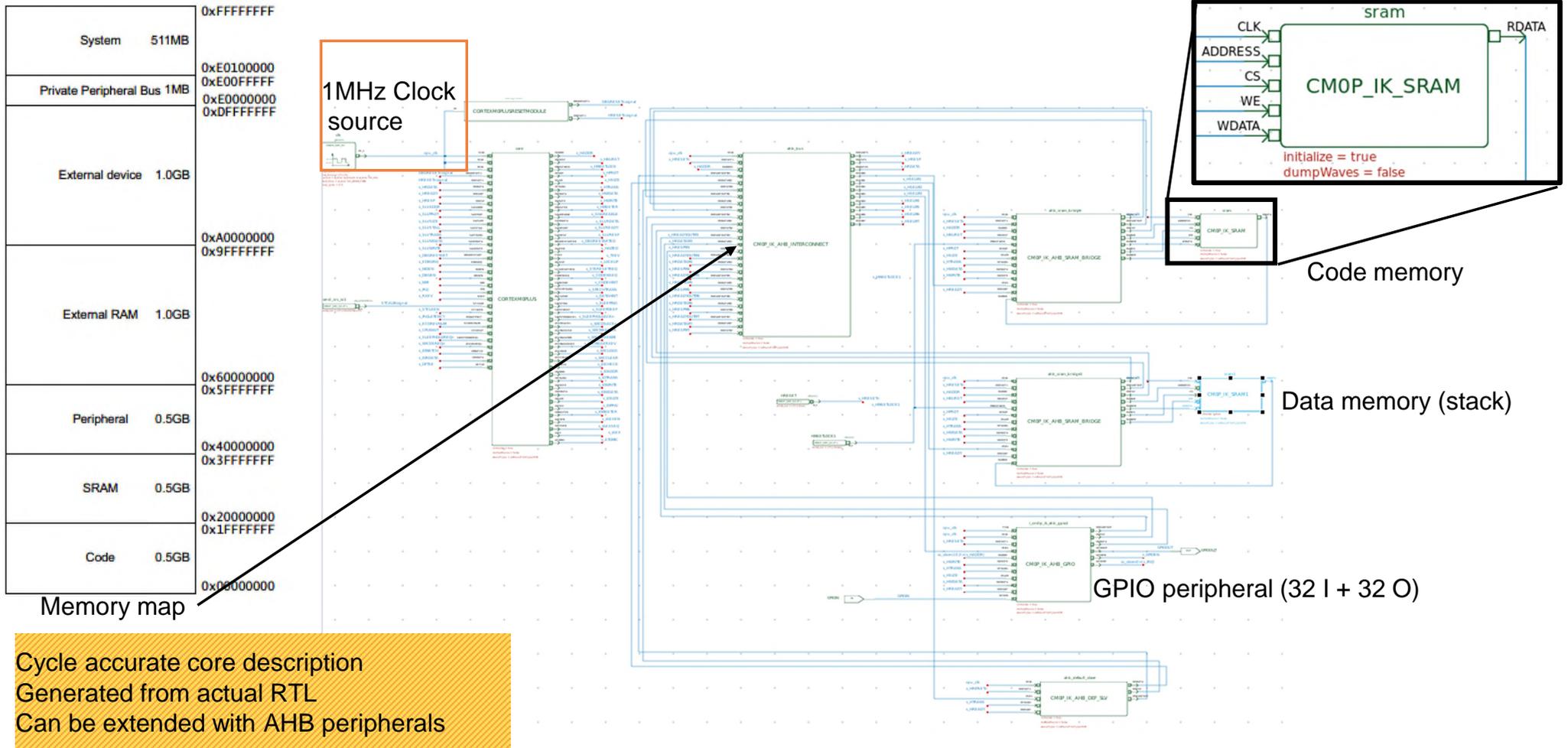


# Table of contents

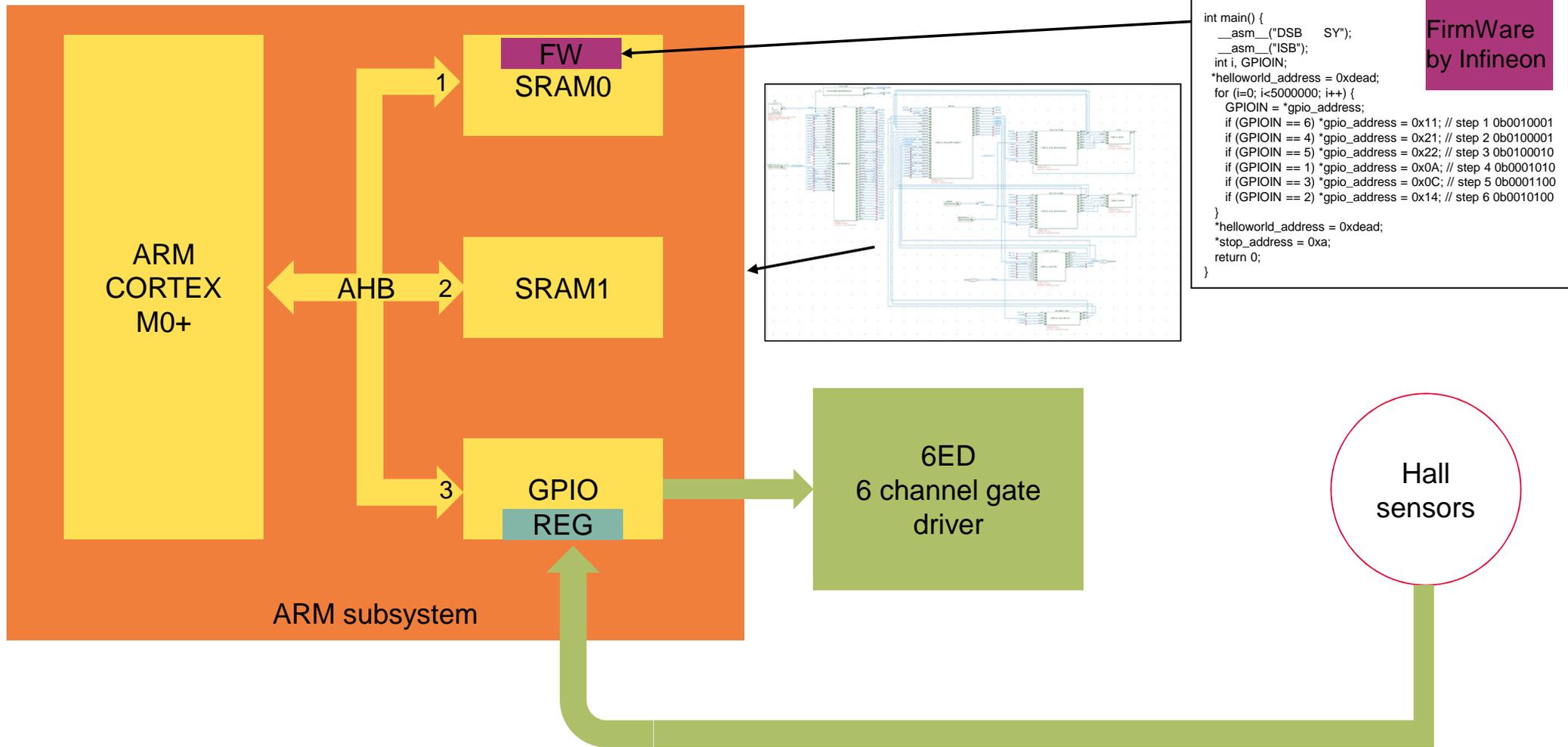
---

|   |                                   |    |
|---|-----------------------------------|----|
| 1 | System introduction               | 3  |
| 2 | Trapezoidal motor characteristics | 6  |
| 3 | Block commutation                 | 10 |
| 4 | ARM Cortex M0+ integration        | 15 |

# Microcontroller subsystem schematics from ARM/COSEDA



# MCU implementation using cortex M0+



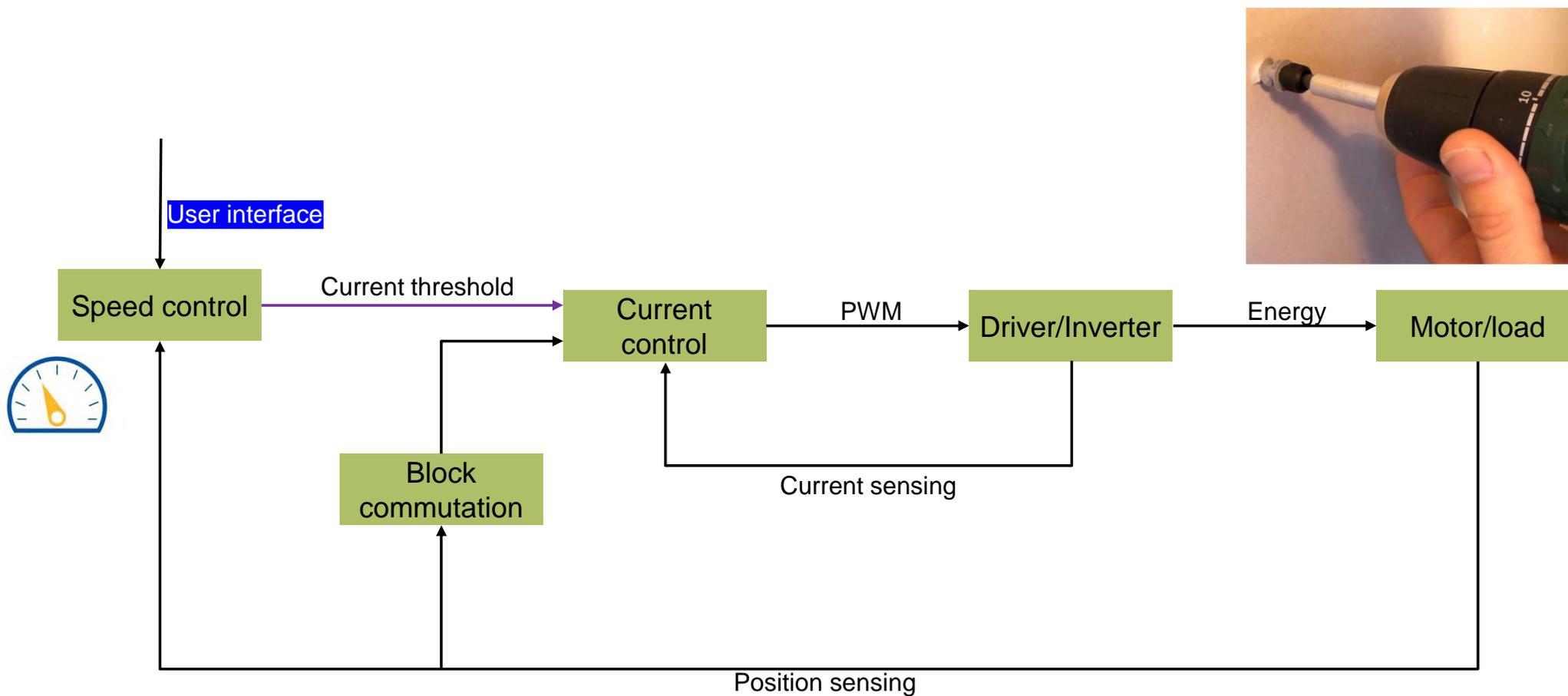
# ARM processor firmware can be edited and compiled in COSIDE

The screenshot displays the COSIDE IDE environment. The main window shows the source code for a project named 'm0\_plus\_motor'. The code includes a main function that configures GPIO pins and writes to a hardware register. The console window at the bottom shows the output of a 'make' command, indicating a successful build of the 'm0\_plus\_motor' project. A green callout box on the right side of the image points to the 'Build Targets' panel, with the text 'Compile code button'.

SW Project and makefile

Compile code button

# Further developments: motor control loop, Space Vector PWM





Part of your life. Part of tomorrow.