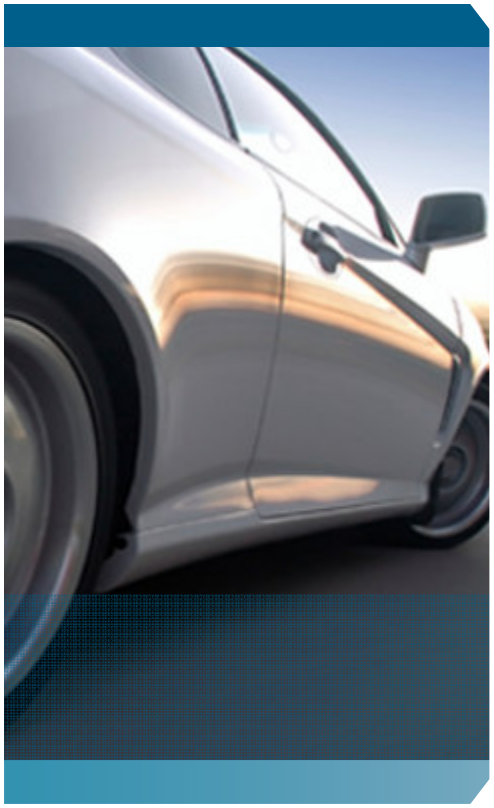


Freescale Technology Forum

Design Innovation.

June, 2008

Hands-on Workshop: Motor Control Part 4 - Brushless DC Motors Made Easy

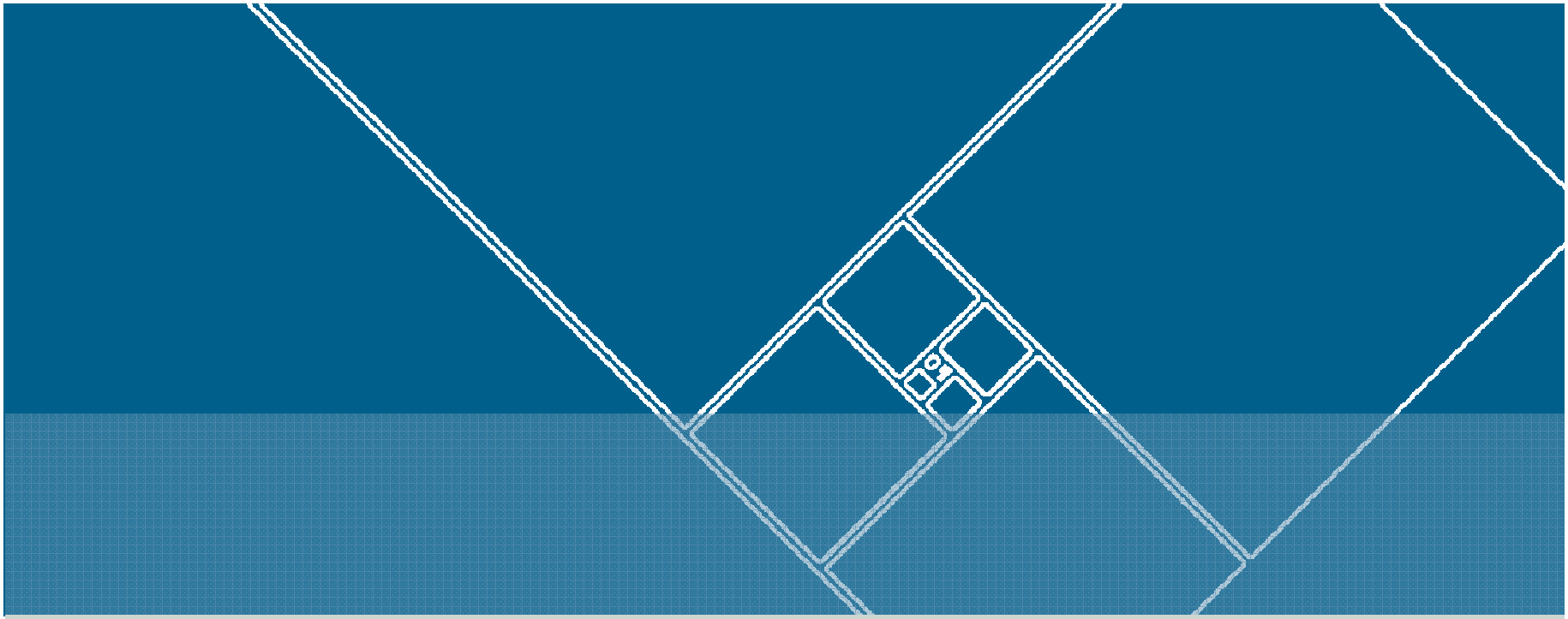
AZ114

Eduardo Viramontes

Applications Engineer

freescale™ semiconductor

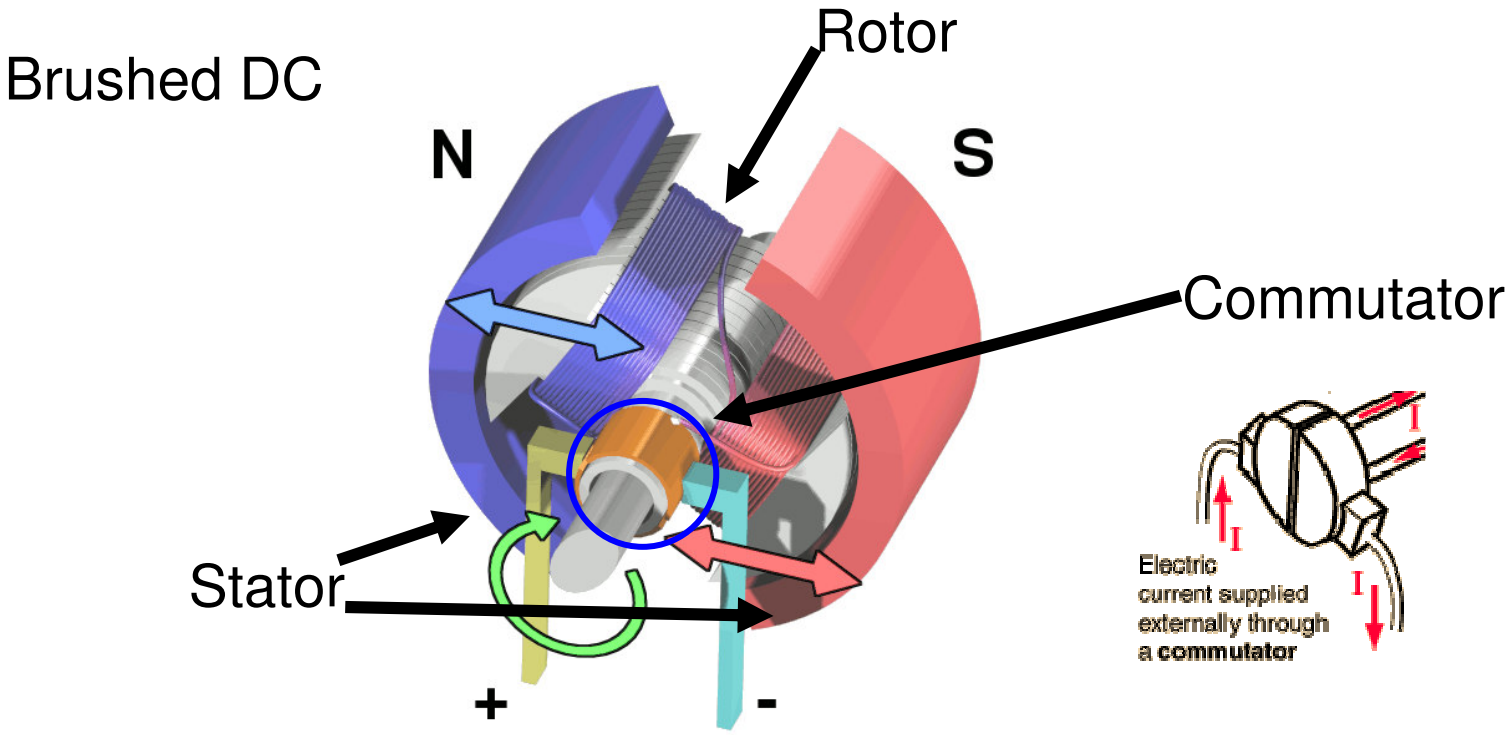
Agenda



Motor Anatomy

freescale™ semiconductor

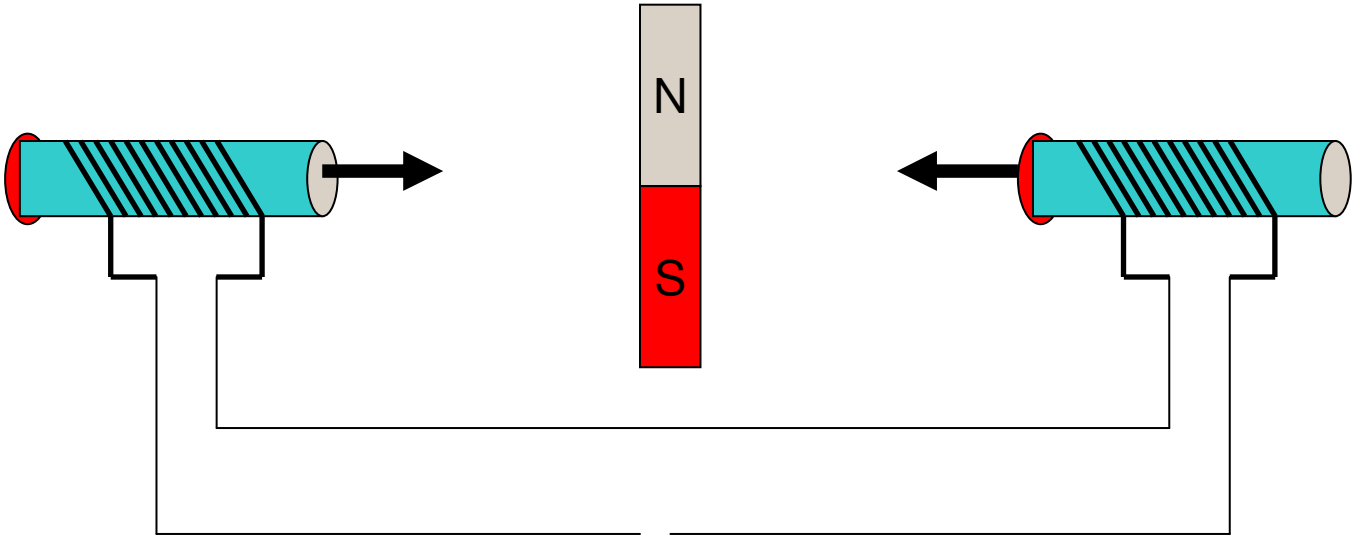
Motor Anatomy



- The first electric motor was the Brushed DC Motor
  - Basic idea is to repel rotor from stator

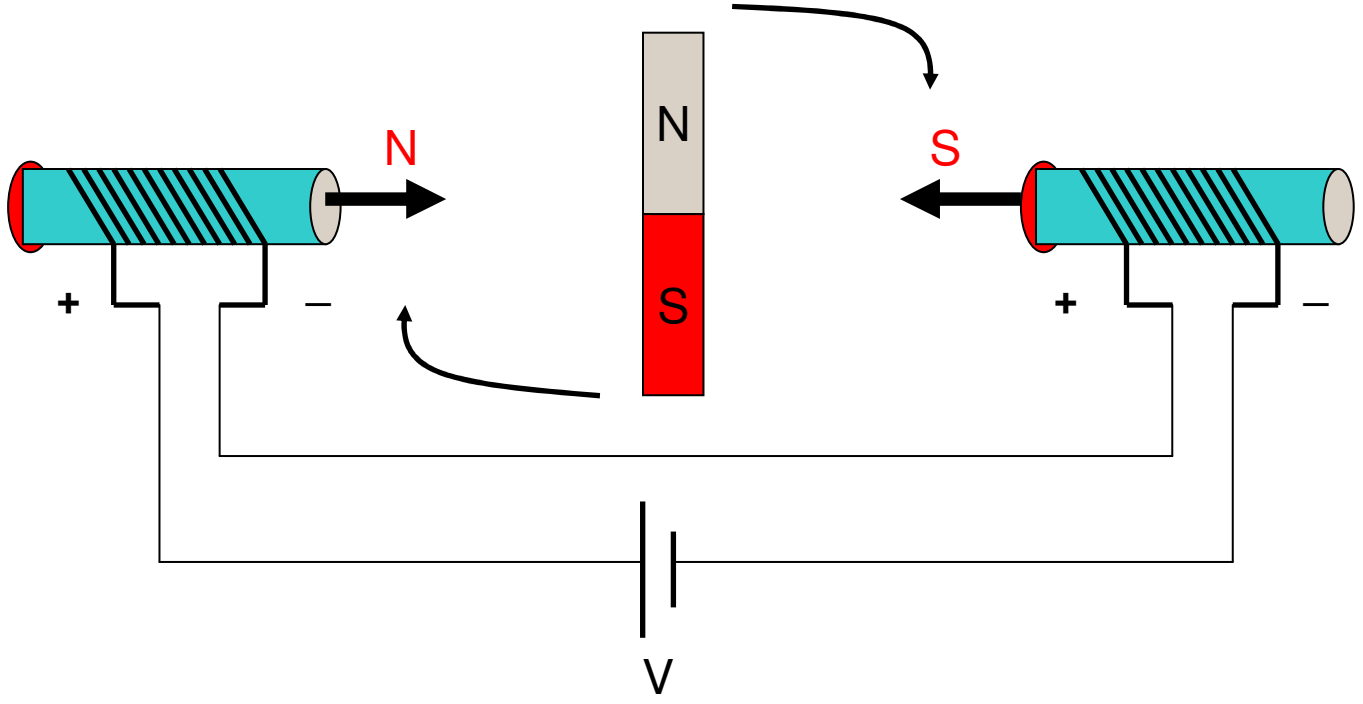
freescale™ semiconductor

Motor Fundamentals



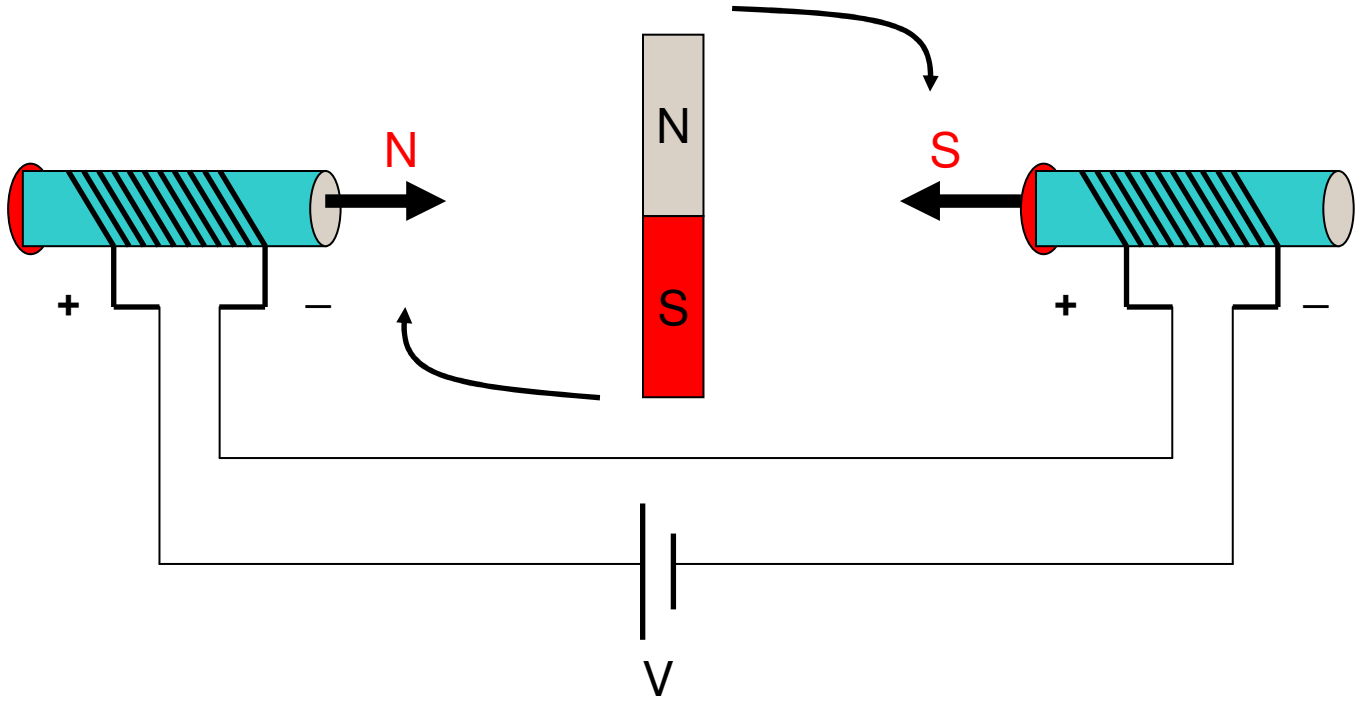
freescale™ semiconductor

Motor Fundamentals



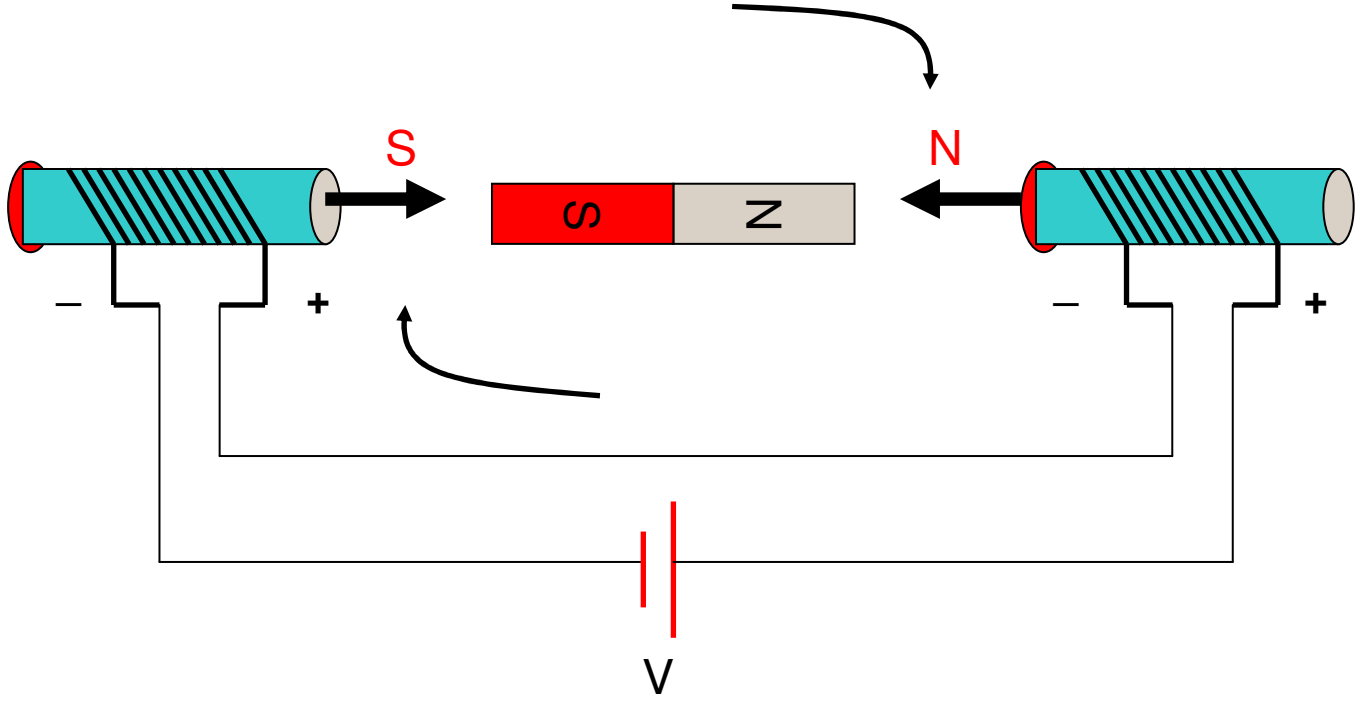
freescale™ semiconductor

Motor Fundamentals



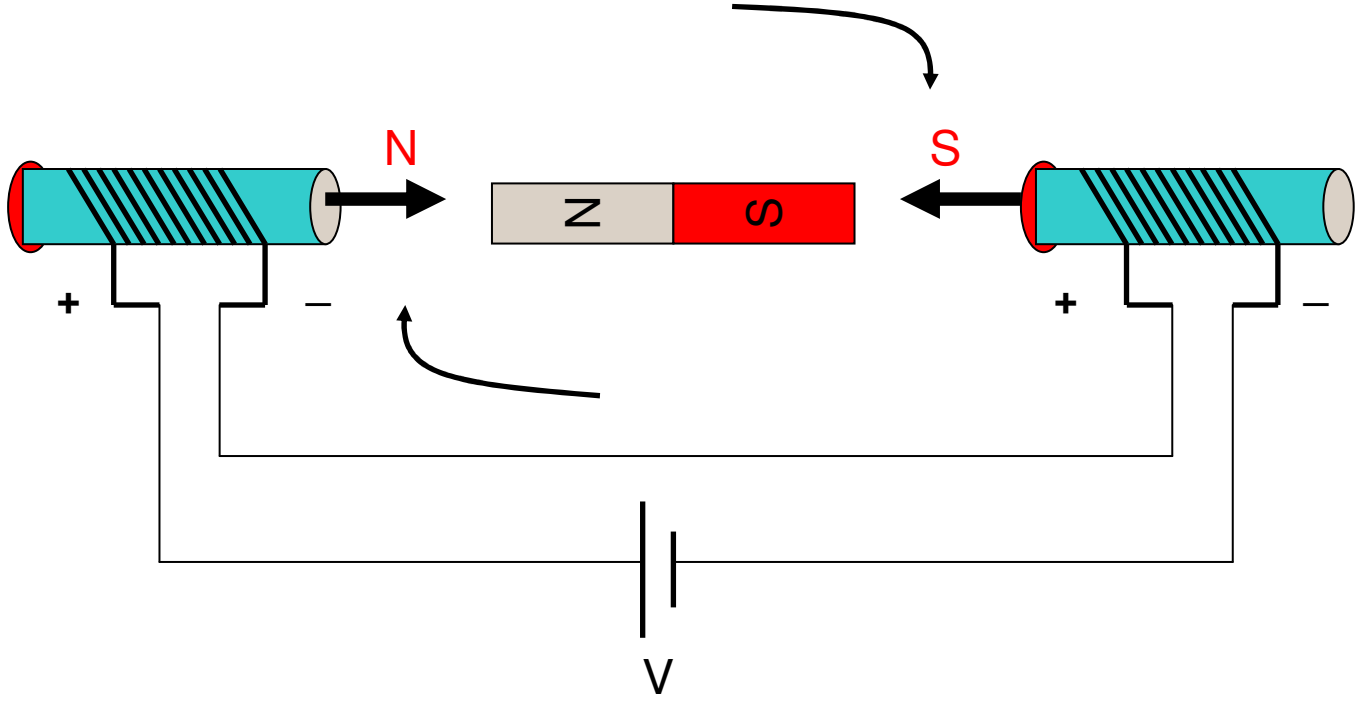
freescale™ semiconductor

Motor Fundamentals

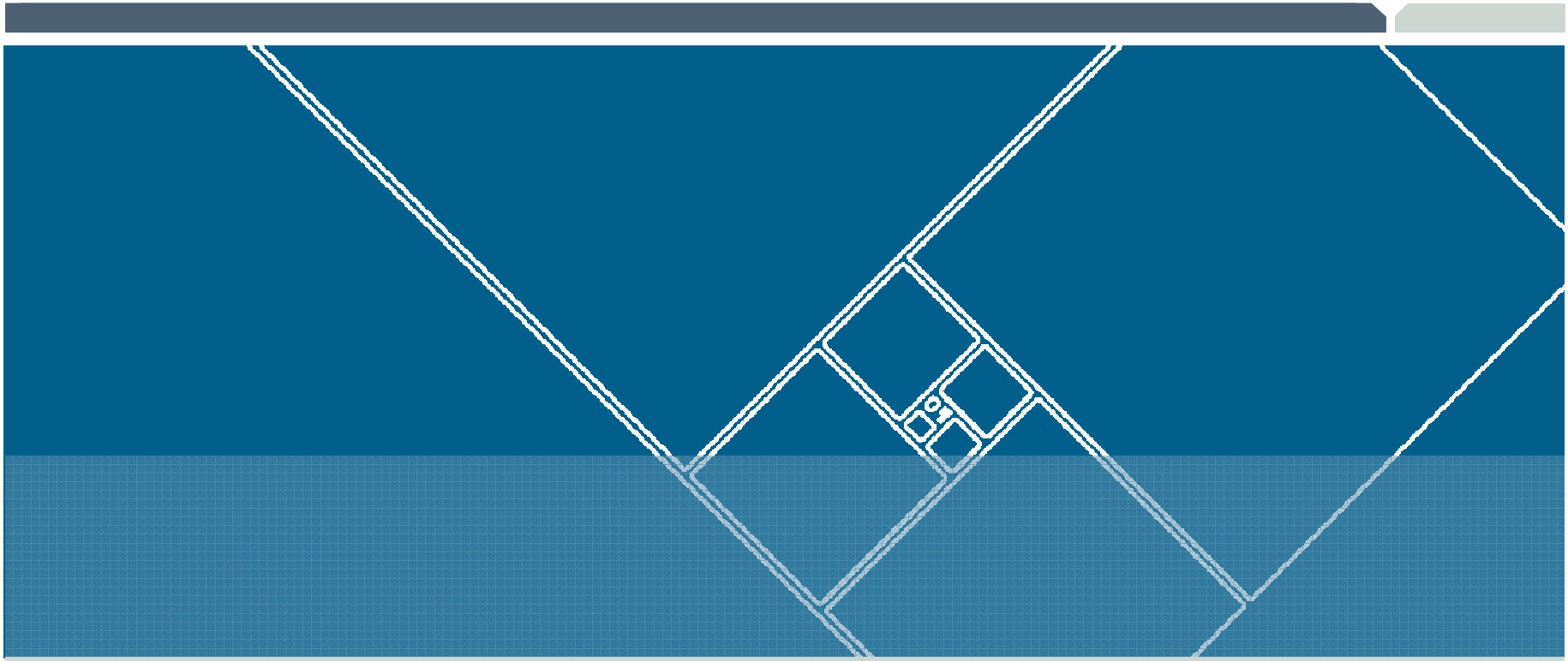


freescale™ semiconductor

Motor Fundamentals



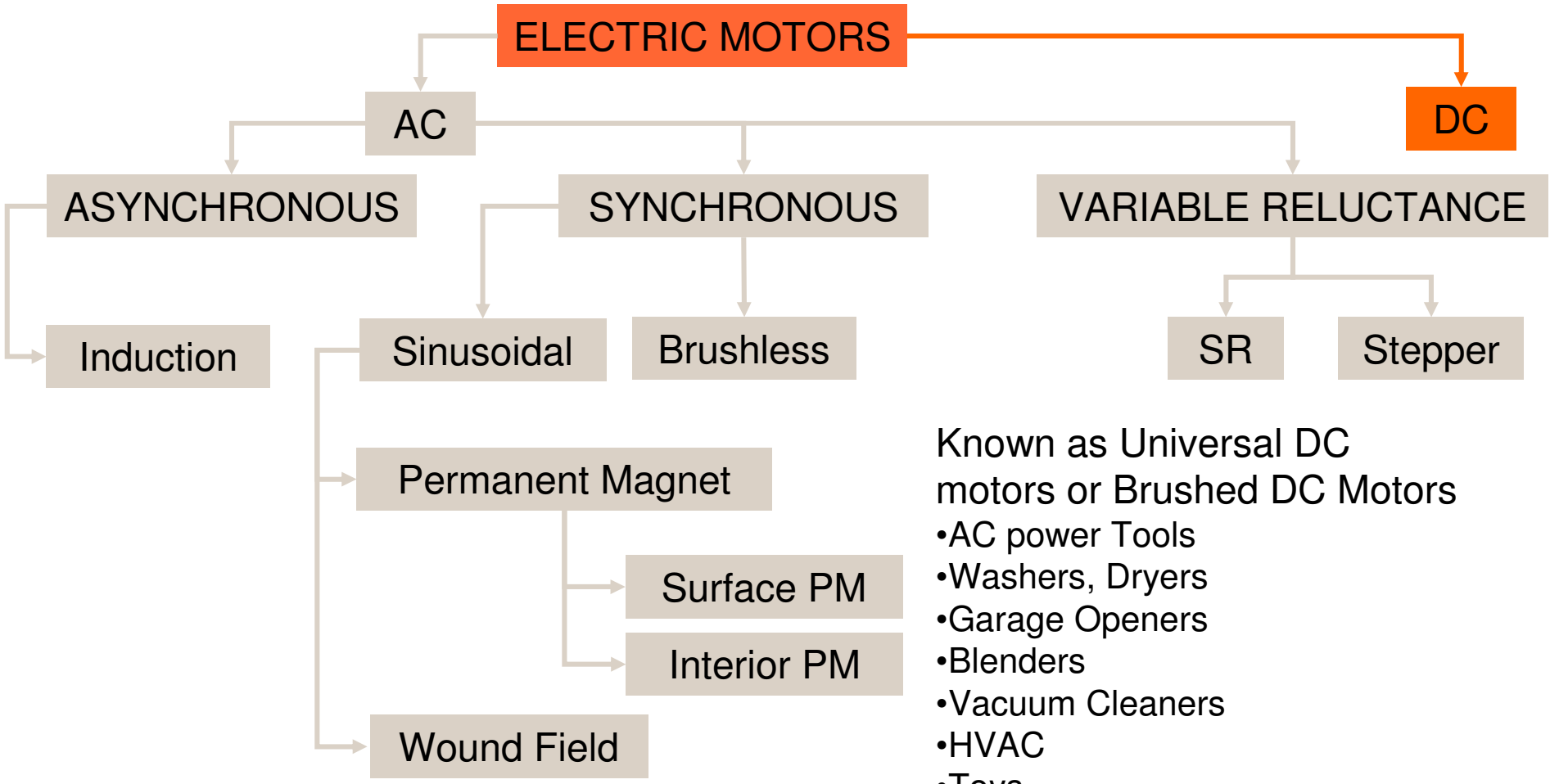
freescale™ semiconductor



Electric Motor Type Classification

freescale™ semiconductor

Electric Motor Type Classification



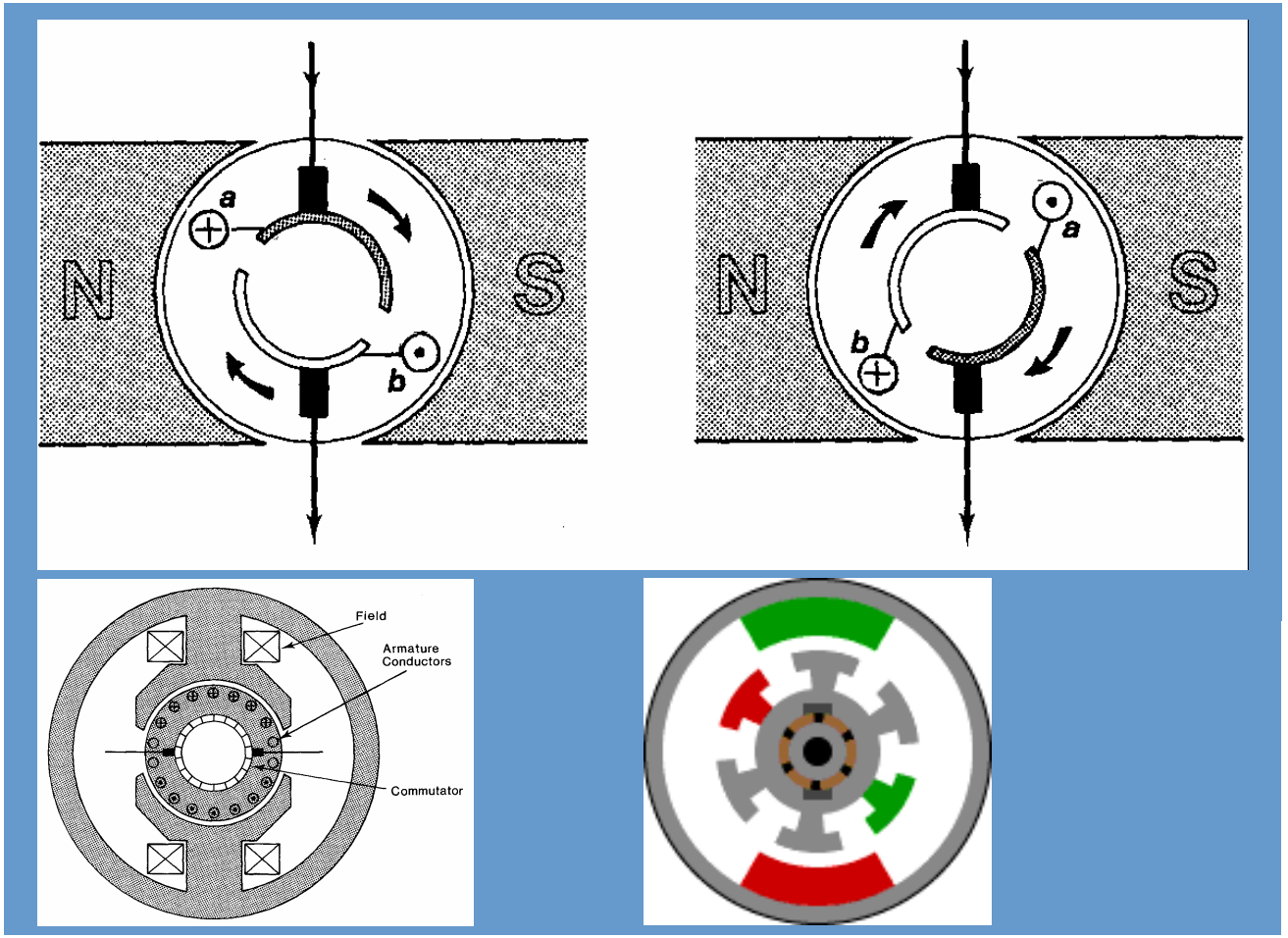
Known as Universal DC motors or Brushed DC Motors

- AC power Tools
- Washers, Dryers
- Garage Openers
- Blenders
- Vacuum Cleaners
- HVAC
- Toys

freescale™ semiconductor

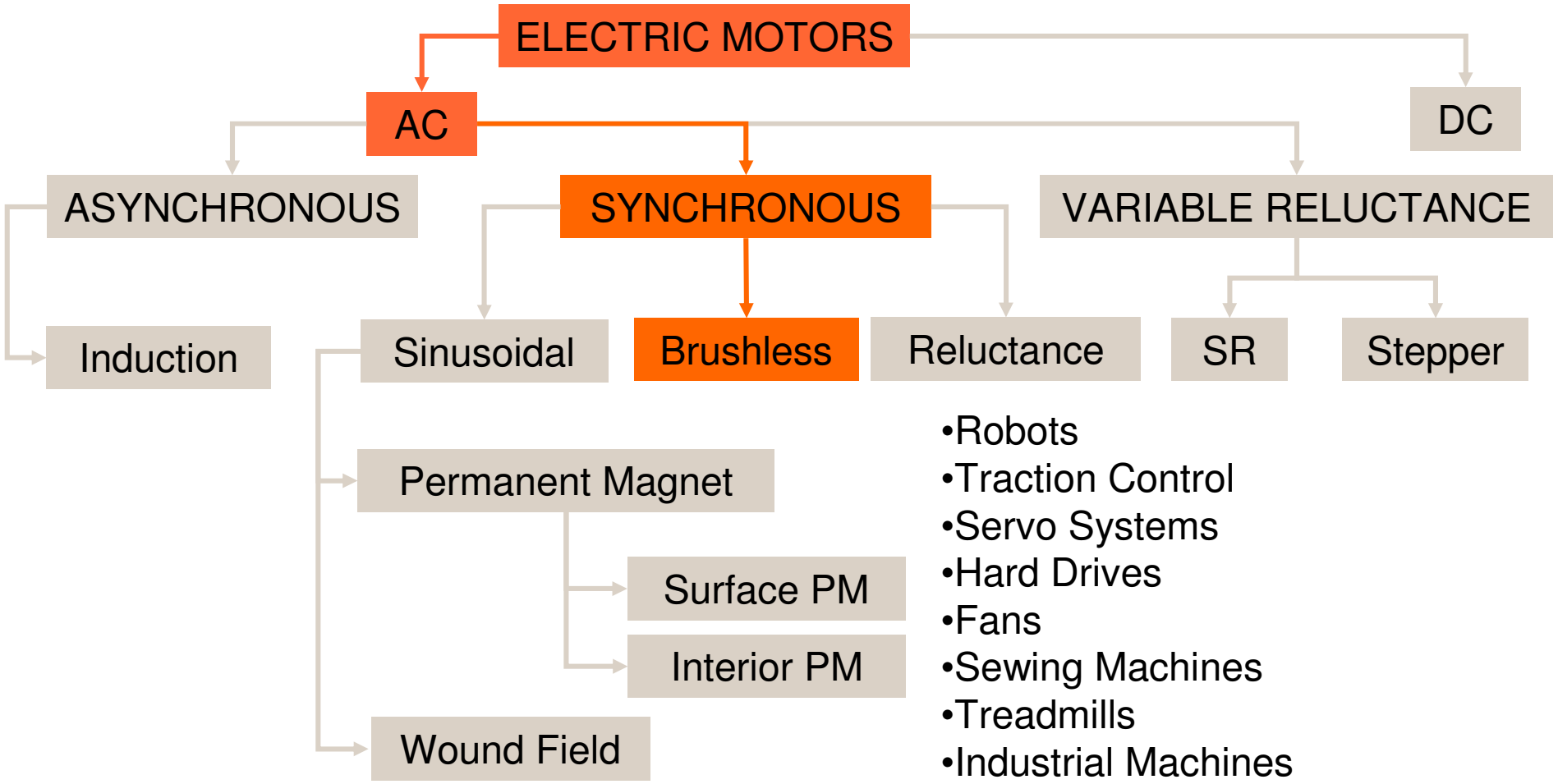
Brushed DC Motors

- Rotation due to electromagnetic force
- Continues rotation with multiple coils
- Undesirable effects due to friction and current reversing



freescale™ semiconductor

Electric Motor Type Classification

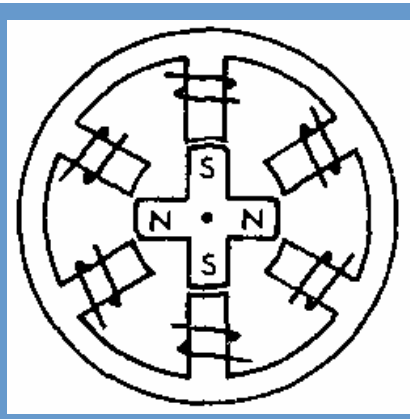


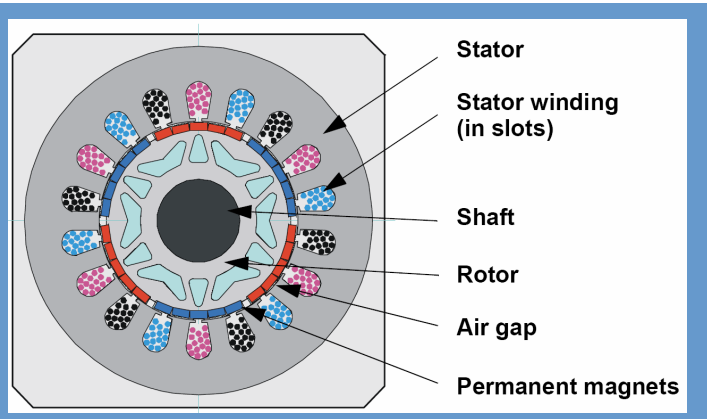
freescale™ semiconductor

Brushless DC (BLDC) Motors

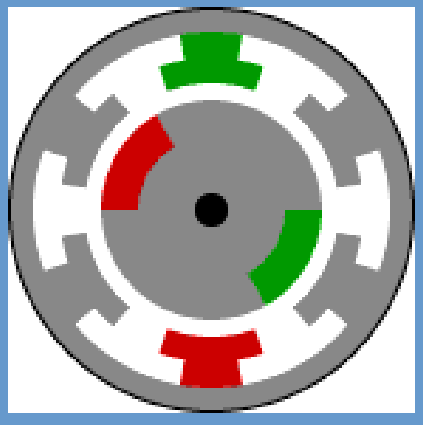
The confusion arises because a BLDC Motor **does NOT** directly operate off a **DC voltage source**

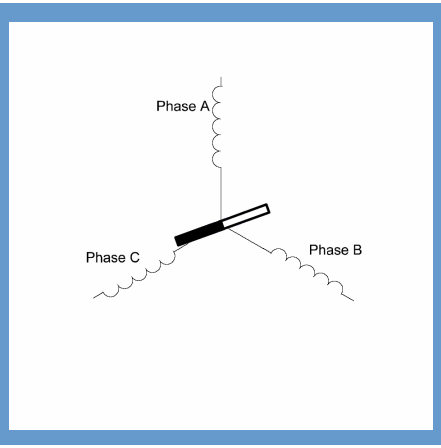
- **Reverse design of brushed motors:**
  - **Magnet is on the rotor**
  - **Inductors are on the stator**





- **Benefits vs. Brushed**
  - **No mechanical commutator**
    (higher speeds)
  - **Better torque/inertia ration**
    (higher acceleration)
  - **Easier to cool**
    (Higher specific outputs)



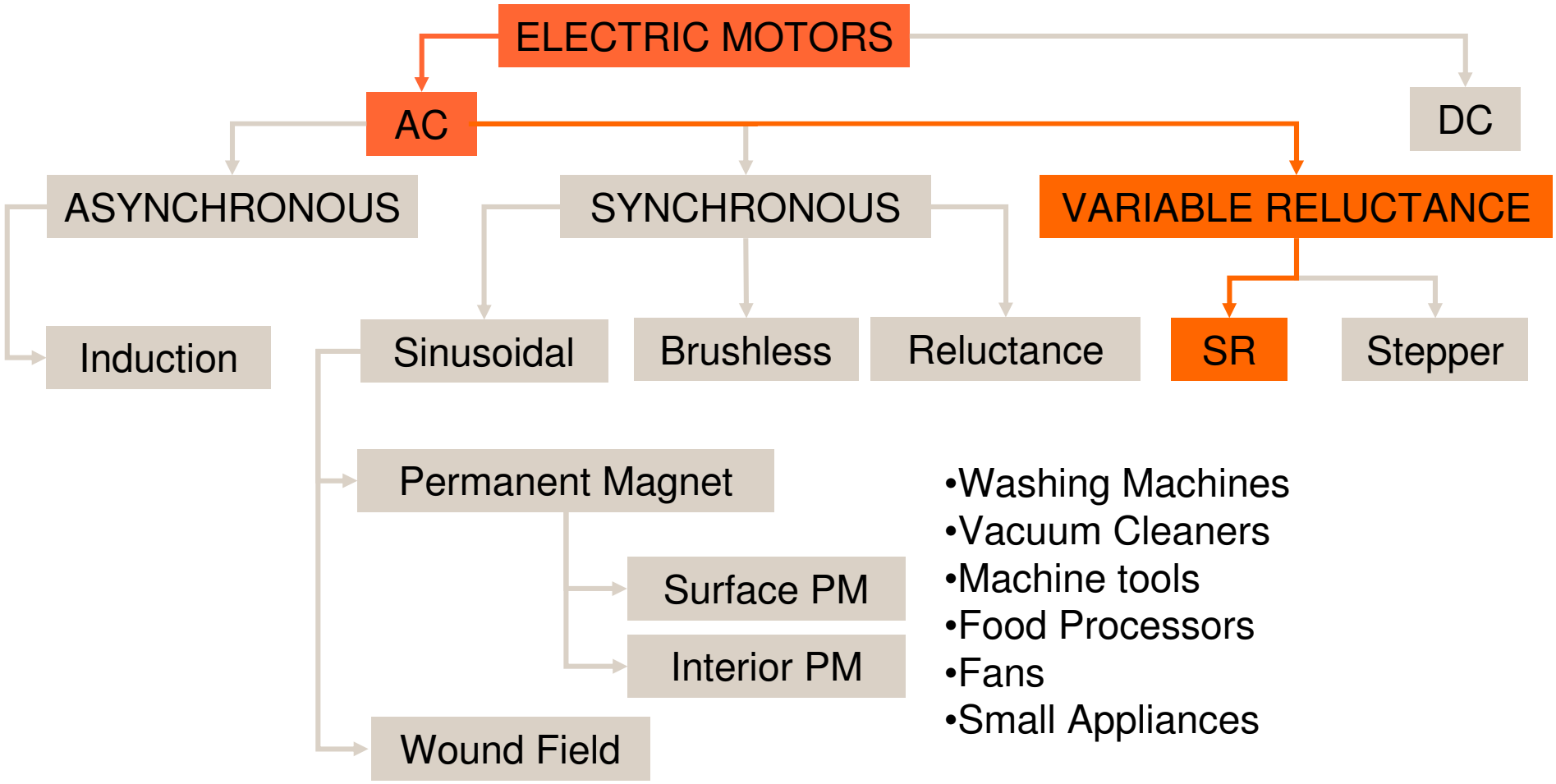


freescale™ semiconductor

Brushed and Brushless Motors Comparison

Feature	Brushed DC motor	BLDC Motor
Commutation	Brushed commutation	Electronic commutation based on Hall position sensors
Maintenance	Periodic maintenance is required	Less required due to absence of brushless
Life	Shorter	Longer
Speed/Torque	Moderately Flat. Higher speeds produces higher friction and this reduces torque.	Flat
Speed range	Lower – Mechanical limitations by the brushes	Higher – No mechanical limitation
Building Cost	Lower	Higher – Permanent magnets
Control	Simple	Complex and expensive
Control Requirements	A controller is required only when variable speed is desired	A controller is always required

Electric Motor Type Classification

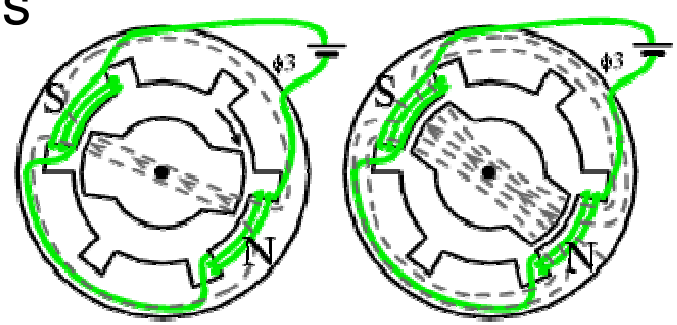


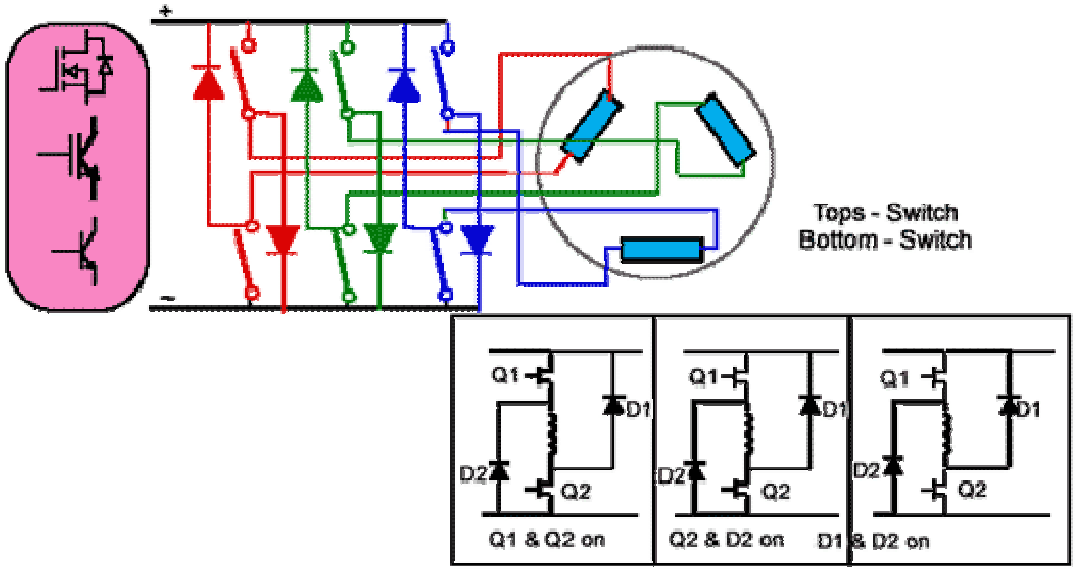
- Washing Machines
- Vacuum Cleaners
- Machine tools
- Food Processors
- Fans
- Small Appliances

freescale™ semiconductor

Switched Reluctance

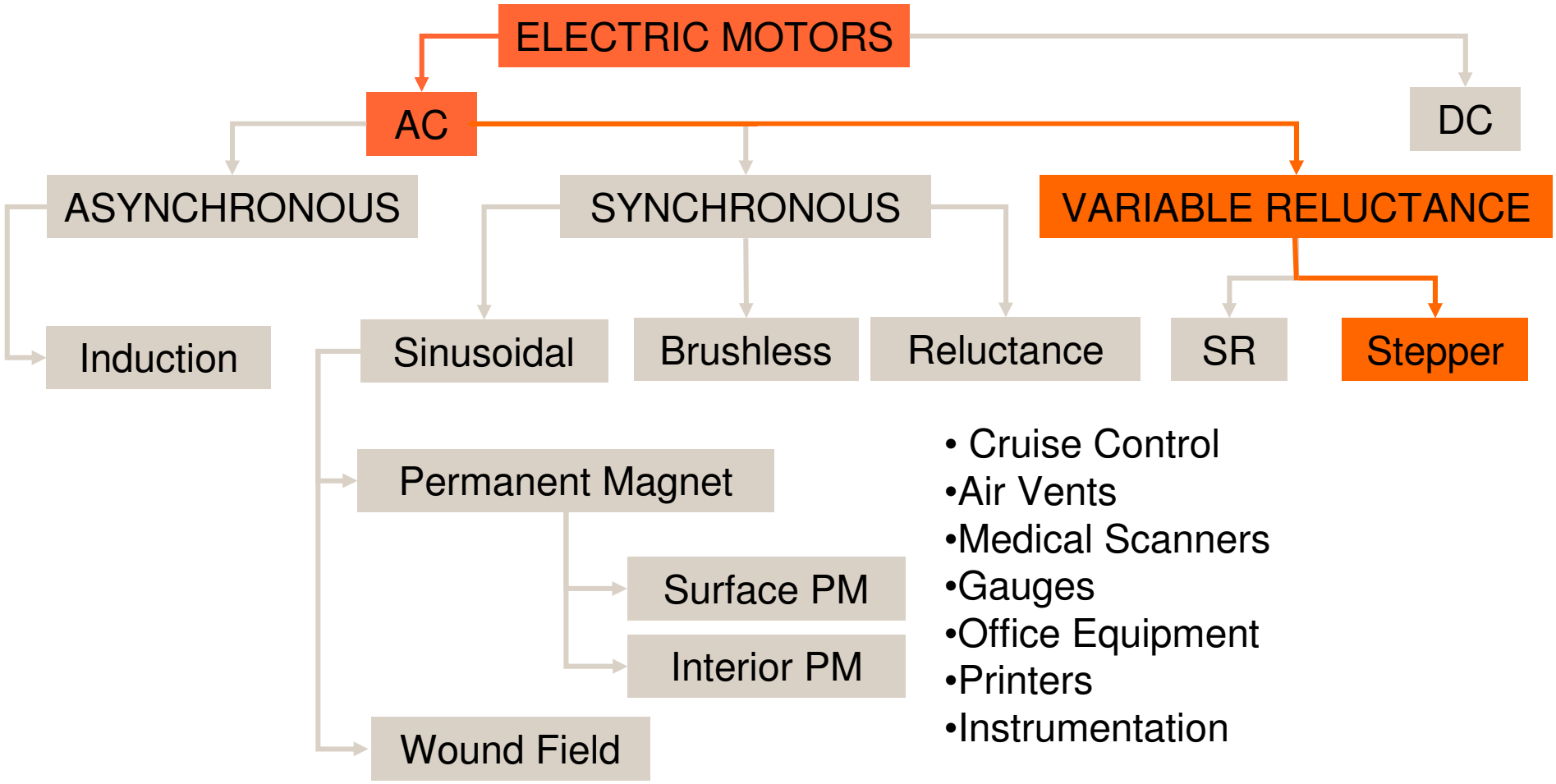
- Both the rotor and stator have salient poles
- The stator winding is comprised of a set
- of coils, each wound to the stator





freescale™ semiconductor

Electric Motor Type Classification

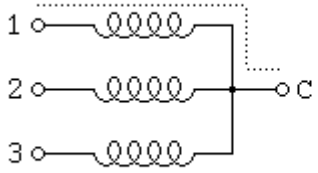


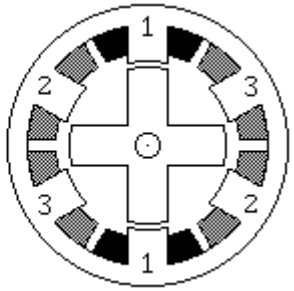
- Cruise Control
- Air Vents
- Medical Scanners
- Gauges
- Office Equipment
- Printers
- Instrumentation

freescale™ semiconductor

Stepper Motor

- These motors turn as different voltages
- are applied to the different windings





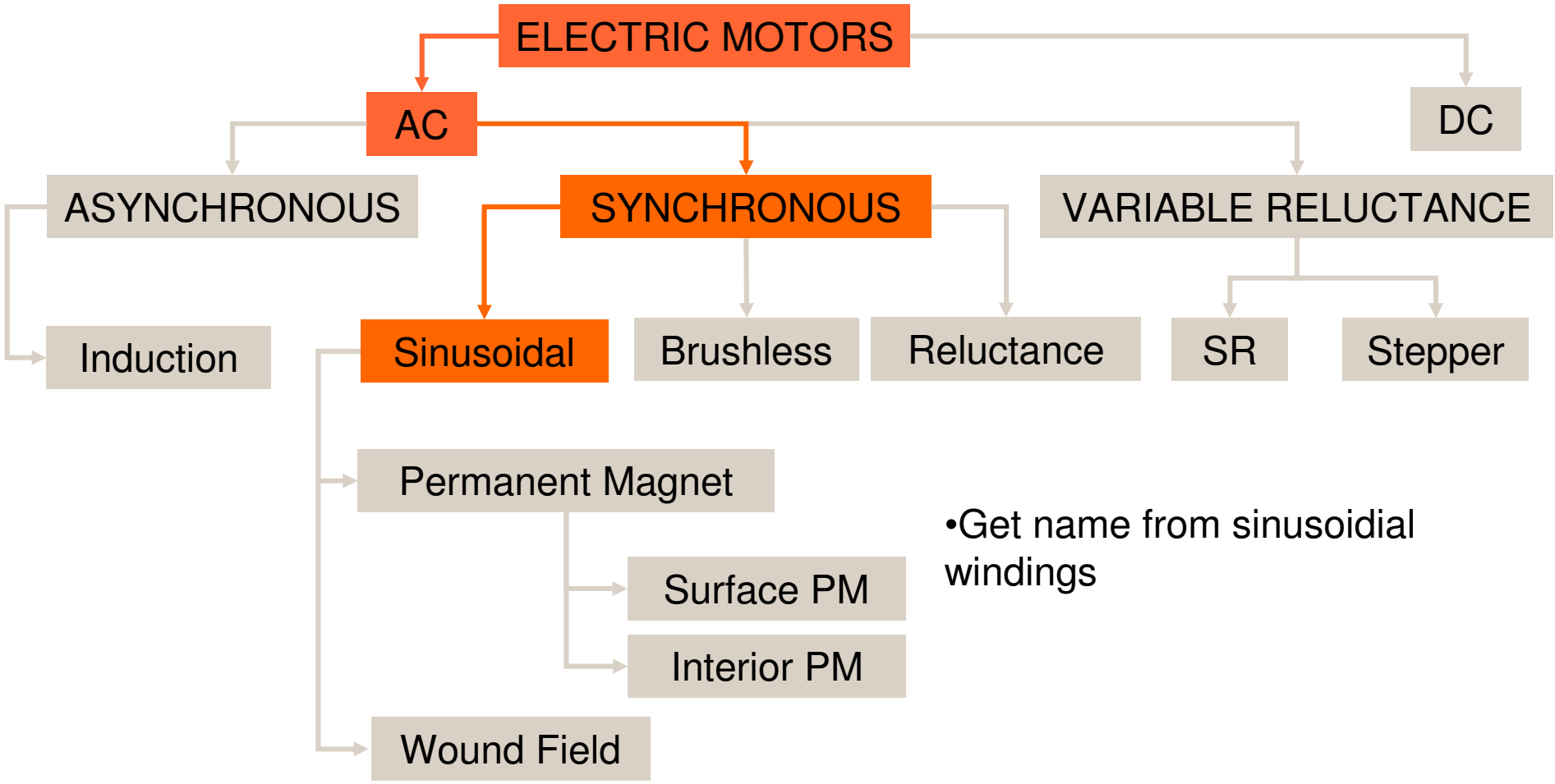
- Field rotates in one direction while rotor
- moves in opposite direction of field

- In this example, field rotates 60° while rotor only moves 30°

- It takes four complete cycles of the control system to rotate motor through one cycle. This is because the Rotor has 4 poles

freescale™ semiconductor

Electric Motor Type Classification

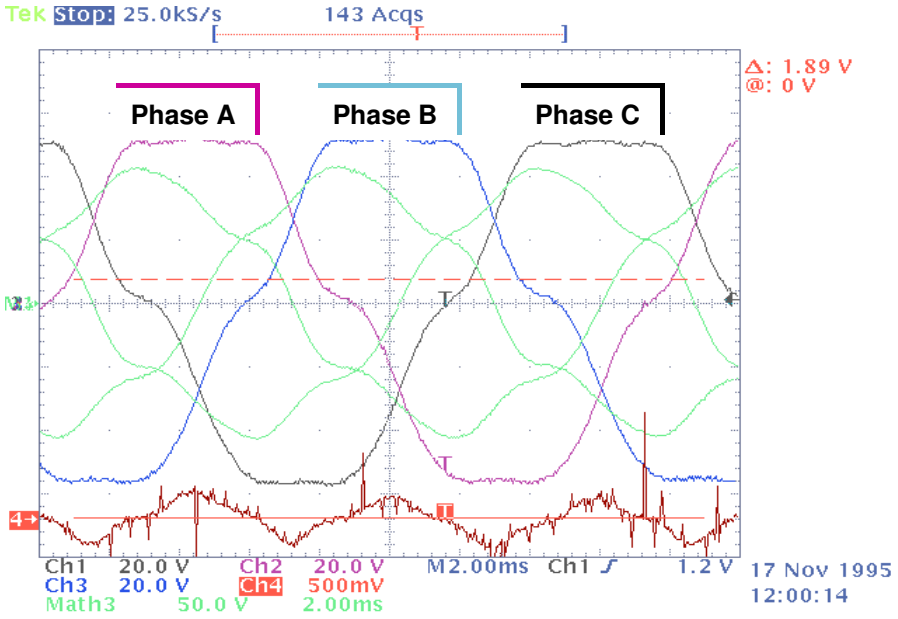


- Get name from sinusoidal windings

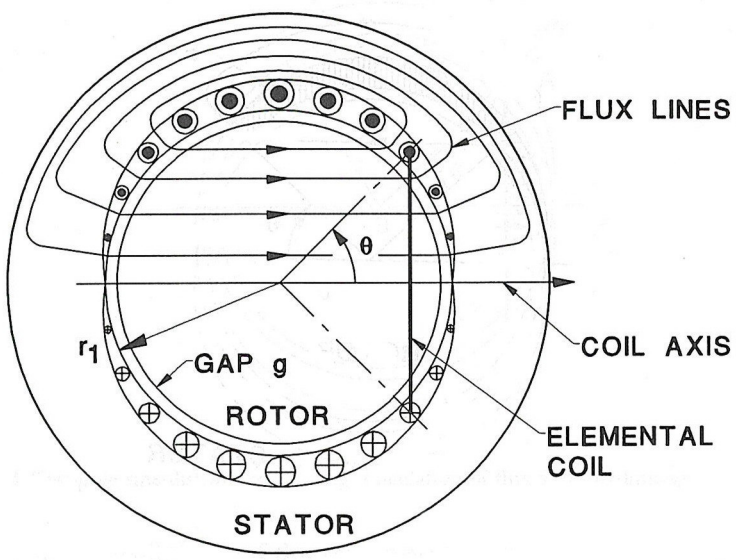
freescale semiconductor

Brushless DC Motor Control

- BLDC Motor versus PMSM Motor
  - Both motors have identical construction. The difference is in stator winding only. The BLDC has distributed stator winding in order to have trapezoidal Back-EMF. The PMSM motor has distributed stator winding in order to have sinusoidal Back-EMF.



Trapezoidal Back-EMF voltage

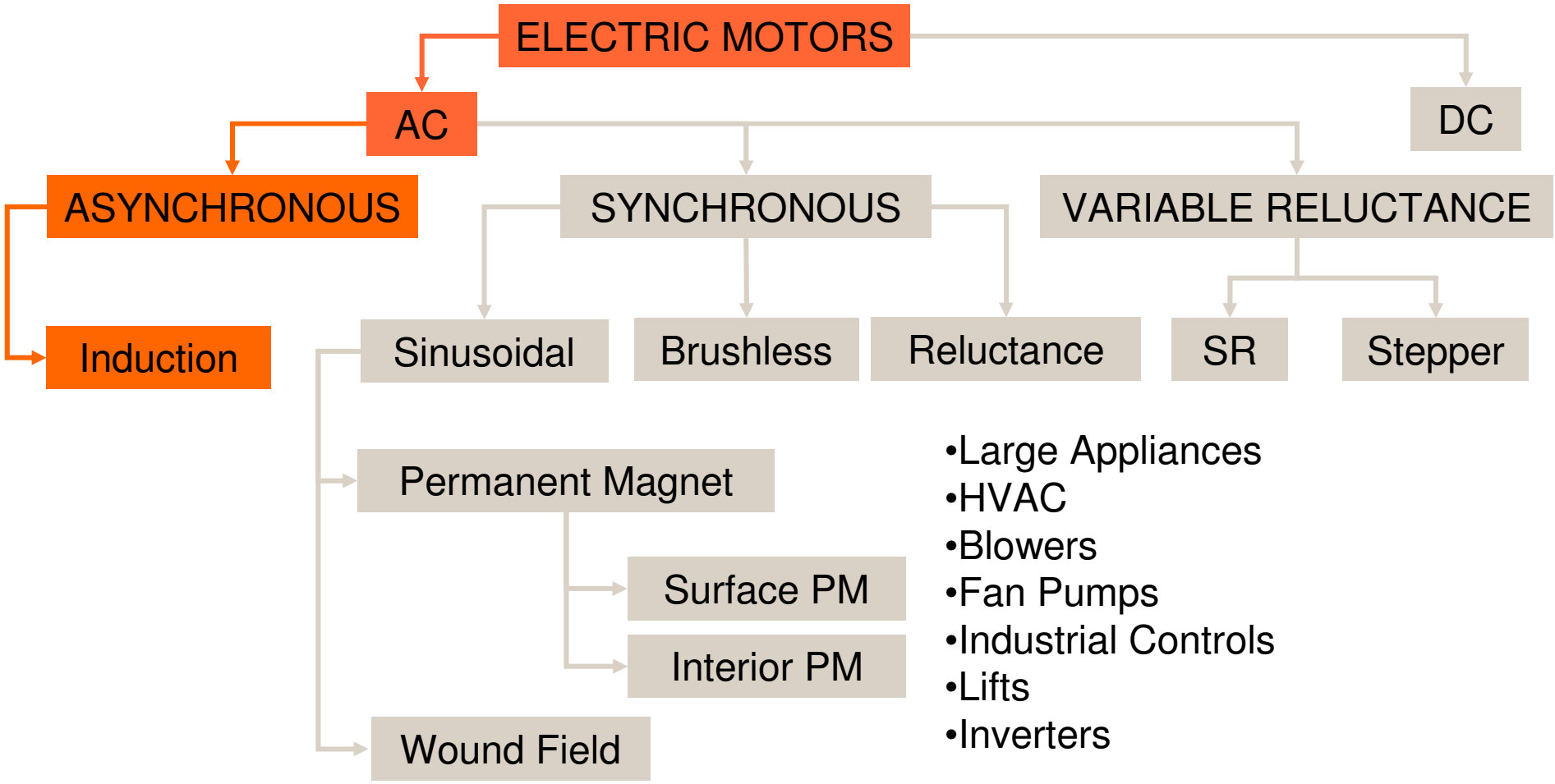


Sinusoidal winding distribution

Source: Hendershot J. R. Jr, Miller TJE: Design of brushless permanent-magnet motors

freescale semiconductor

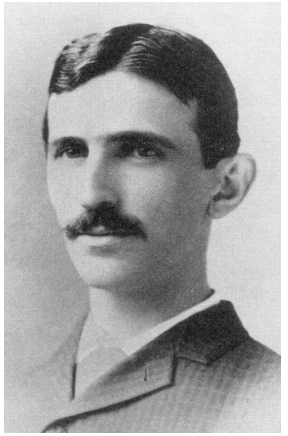
Electric Motor Type Classification



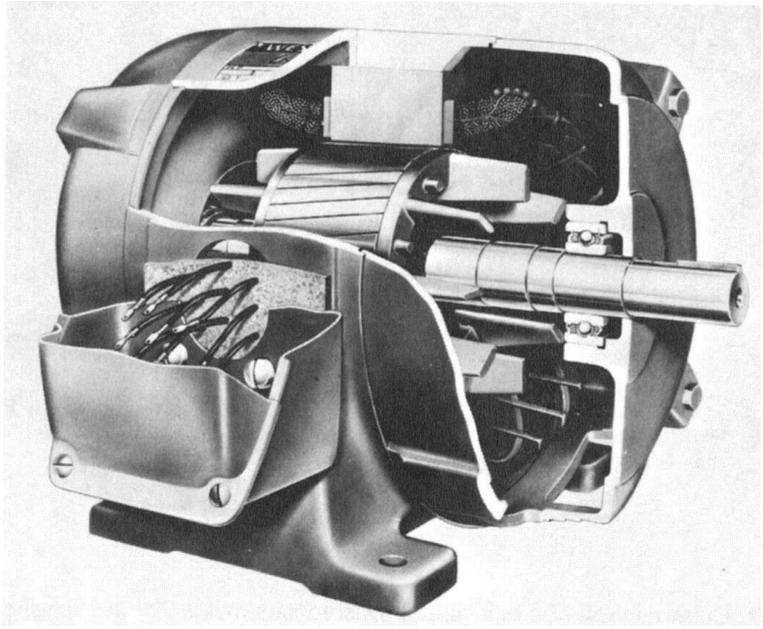
freescale™ semiconductor

Induction Machines

Invented over a century ago by Nikola Tesla



- Stator same as BLDC
- Difference in rotor construction



If properly controlled
- Provides constant torque
- Low torque ripple

No permanent magnets

Think of it as a rotating transformer.

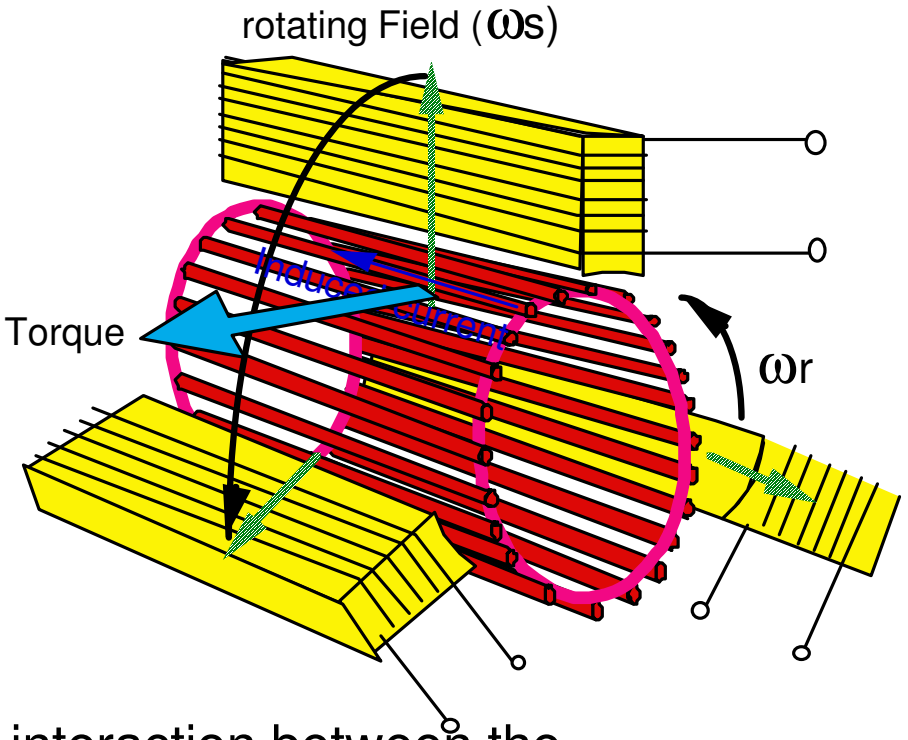
- Stator is the primary
- Rotor is the secondary

Rotor current is “induced” from stator current

freescale™ semiconductor

AC Induction Motor Slip

- **Basic Principle**:
  - The stator is a classic three-phase stator with the winding displaced by 120°
  - The rotor is a squirrel cage rotor in which bars are shorted together at both ends of the rotor by cast aluminum end rings
  - The rotor currents are induced by stator magnetic field.

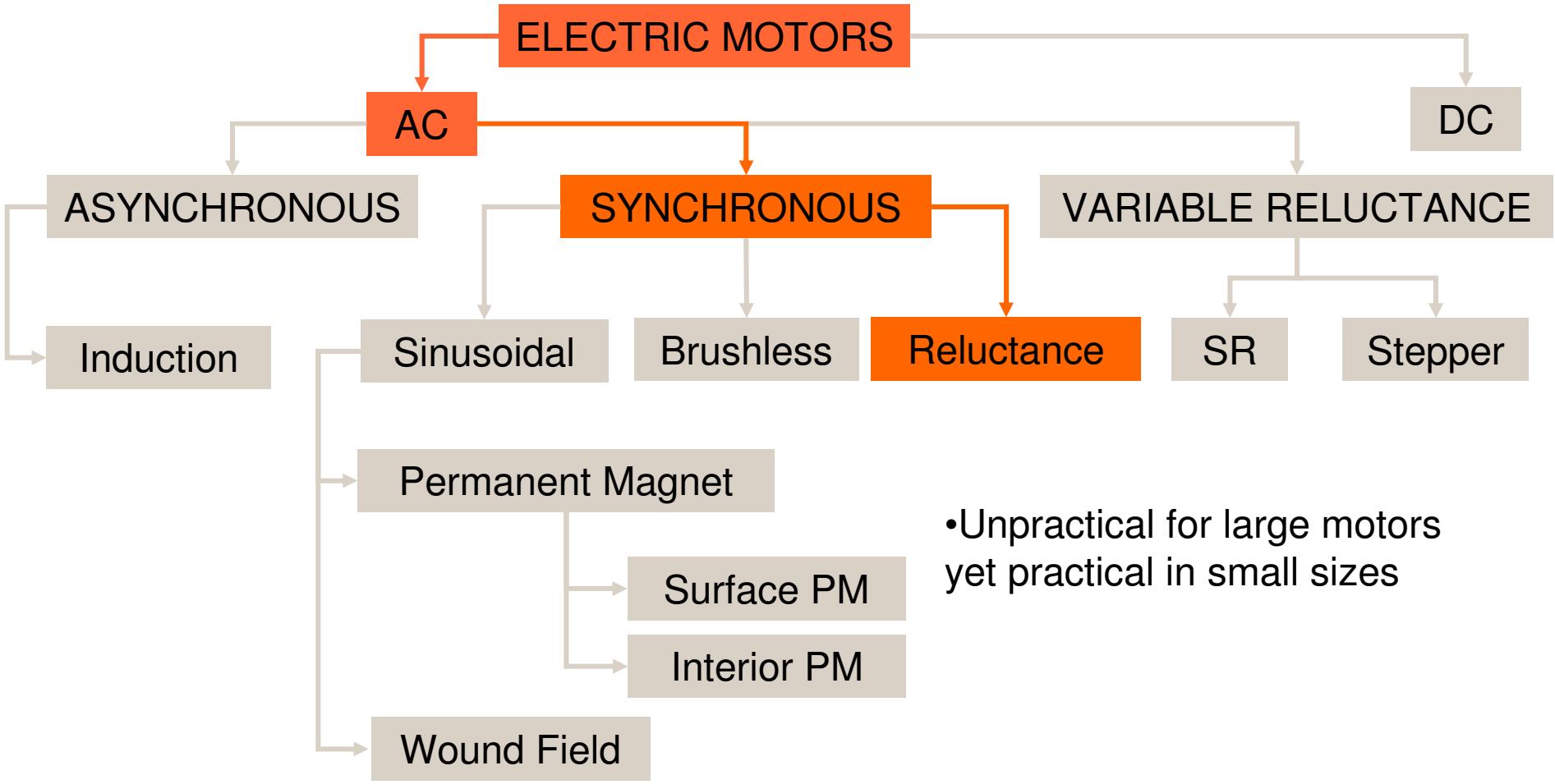


The motor torque is generated by an interaction between the stator magnetic field and induced rotor magnetic field

NO BRUSHES, NO PERMANENT MAGNETS

freescale™ semiconductor

Electric Motor Type Classification

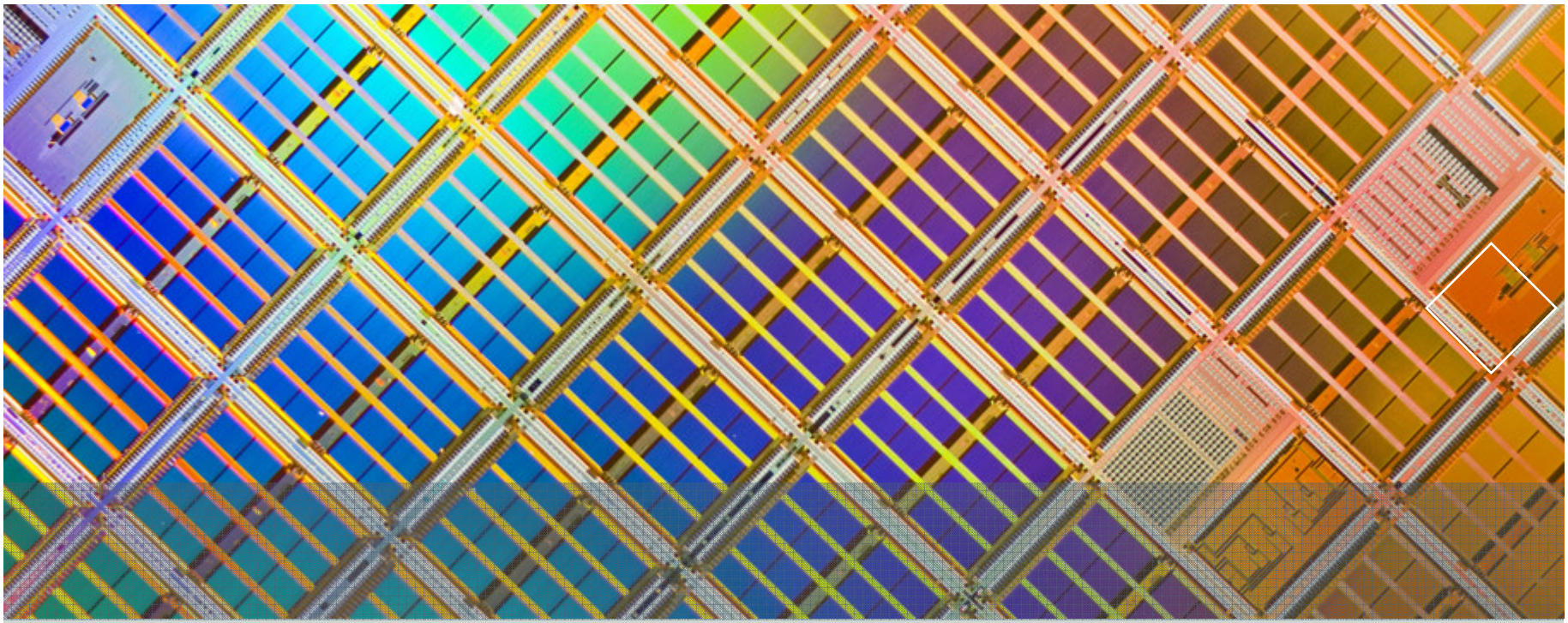


- Unpractical for large motors yet practical in small sizes

freescale™ semiconductor

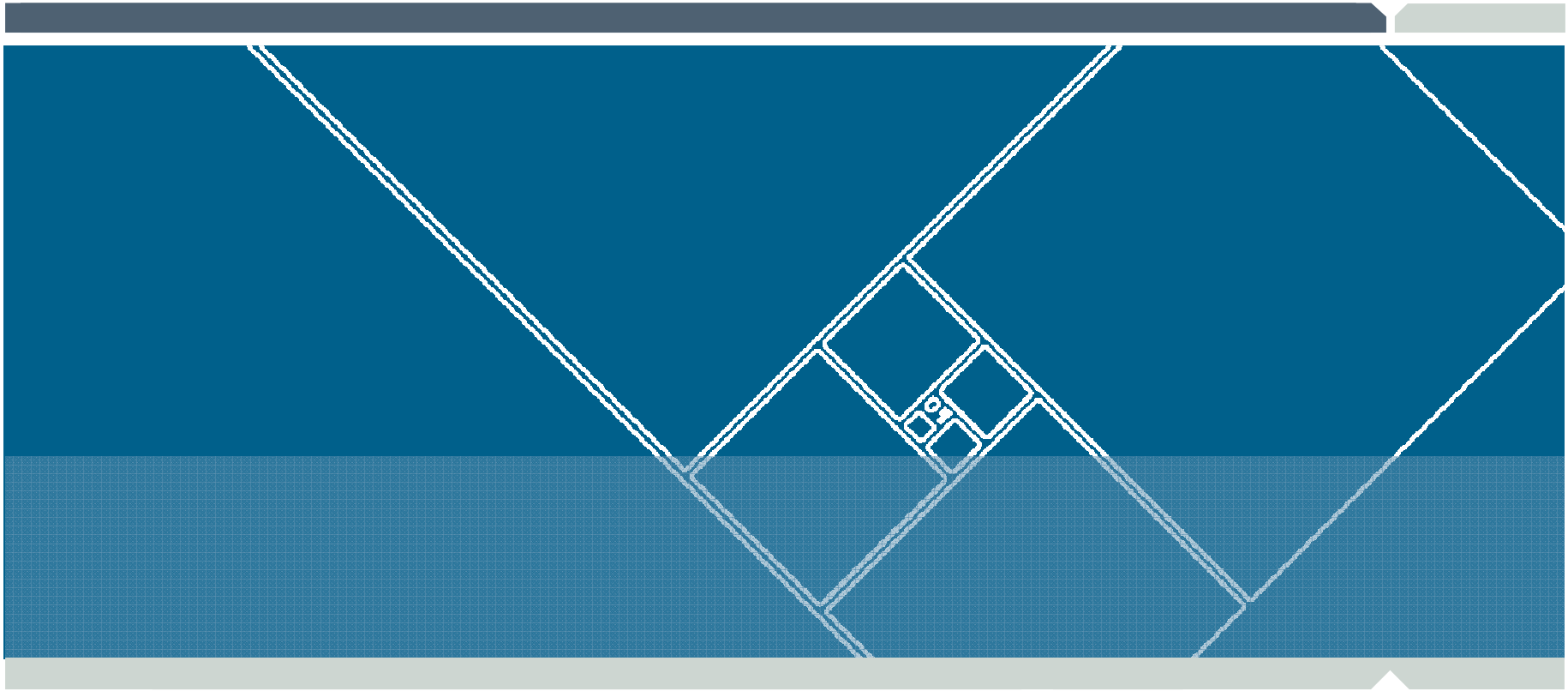
Reluctance

- If the rotating field of a motor is de-energized, it will still develop 10 or 15% of synchronous torque
- If slots are cut into the conductor-less rotor of an induction motor, corresponding to the stator slots, a synchronous reluctance motor results
- Starts like an induction motor but runs with a small amount of synchronous torque



Freescale Roadmap for Motor Control

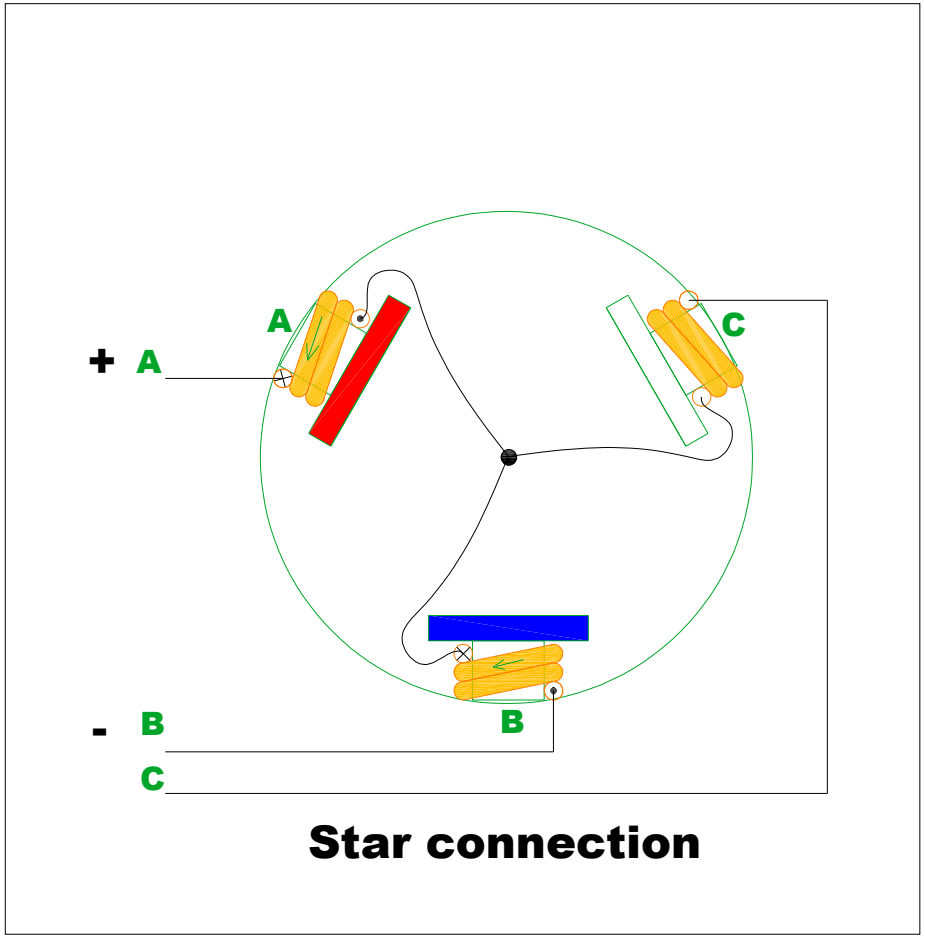
freescale™
semiconductor

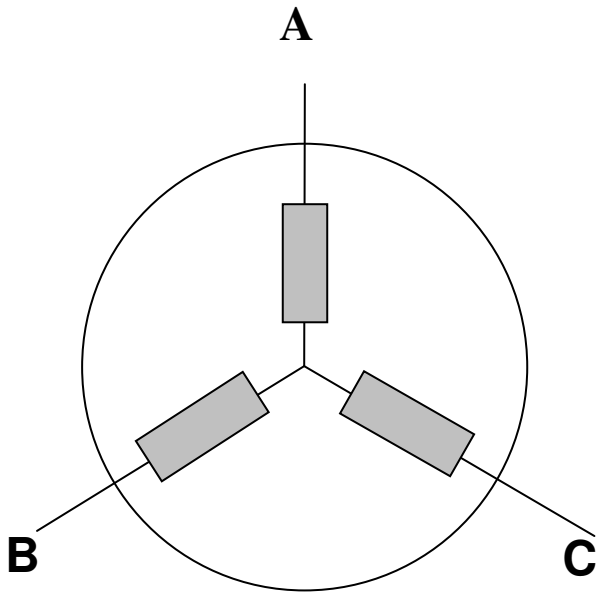


BLDC In Depth:
BLDC Motor Configurations

freescale™
semiconductor

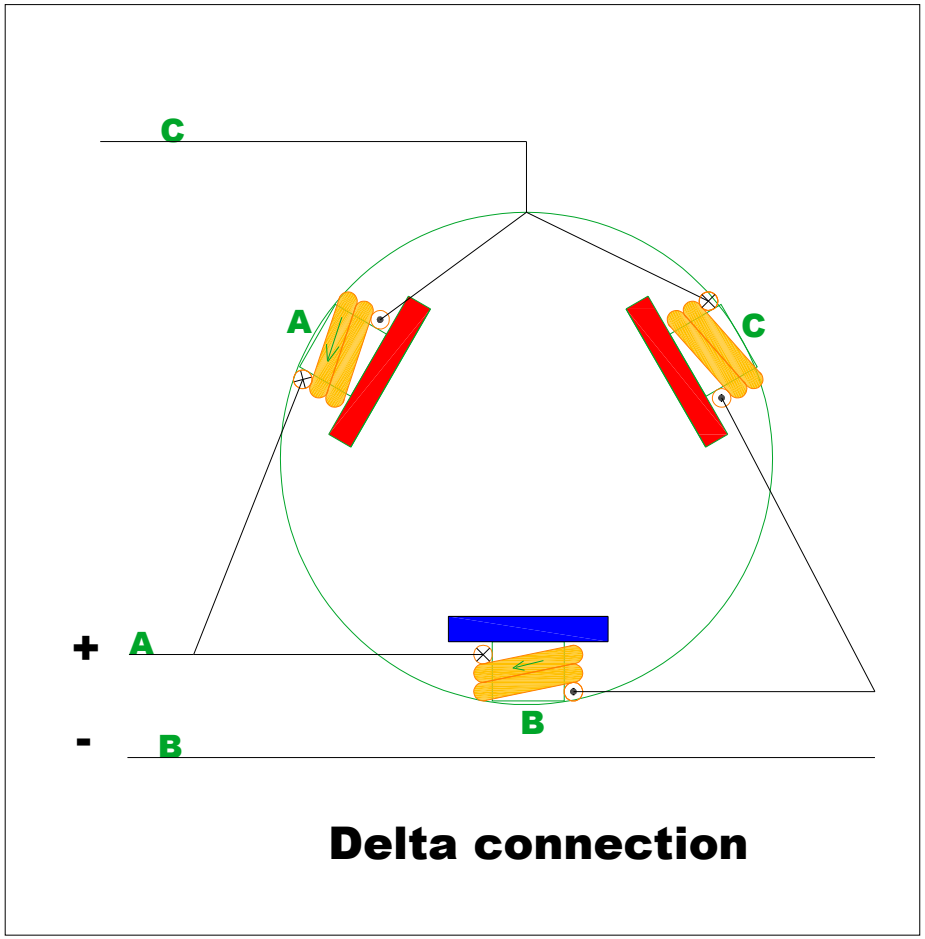
Windings Electrical Connection - Star

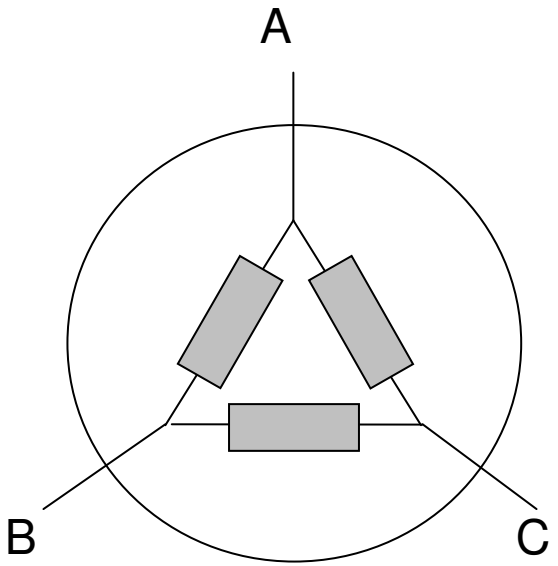




freescale™ semiconductor

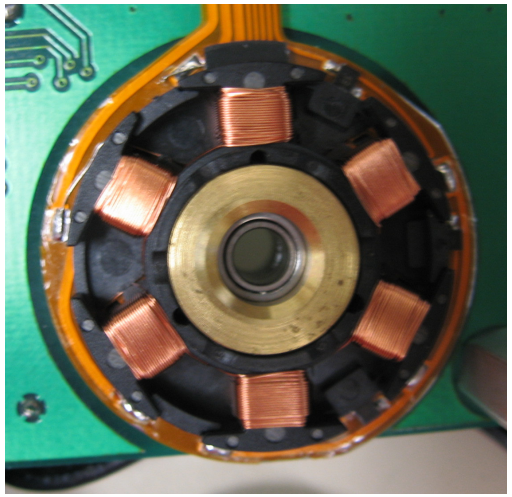
Windings Electrical Connection - Delta

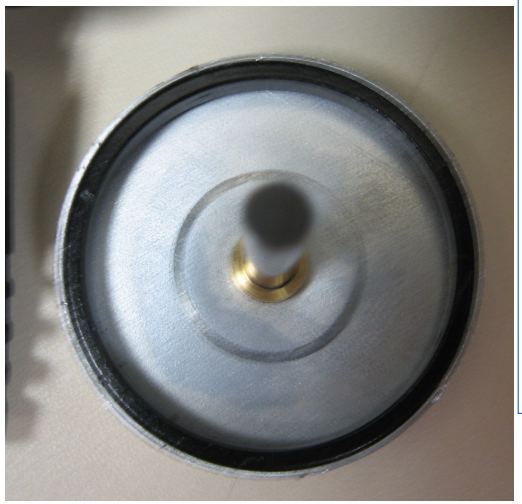


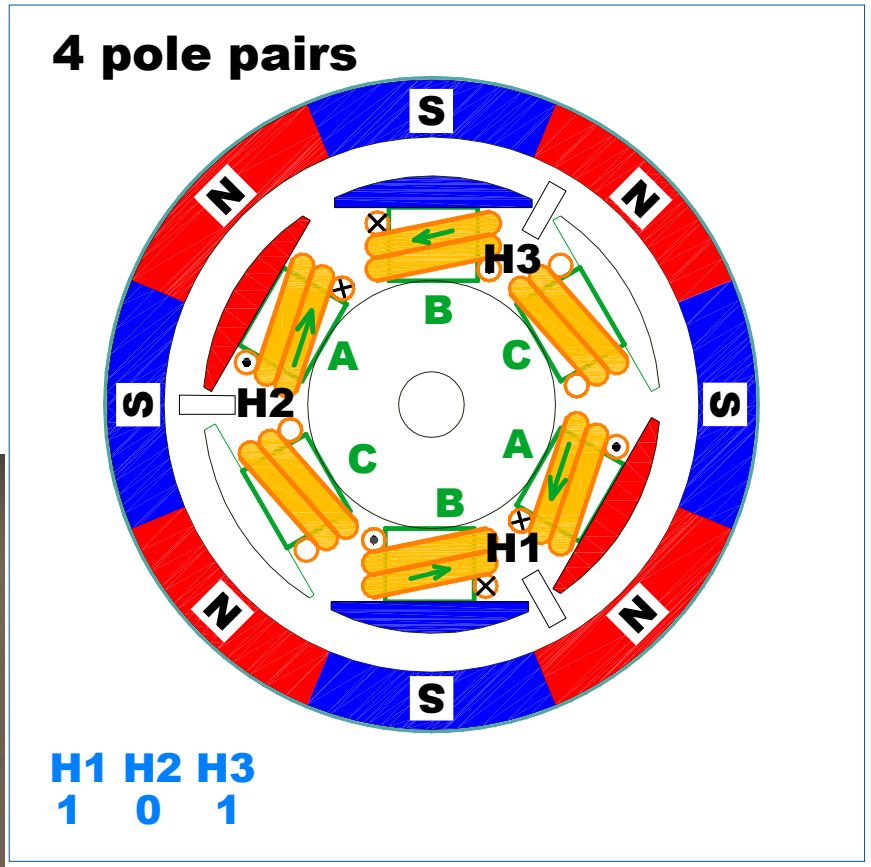


freescale™ semiconductor

BLDC motor configuration

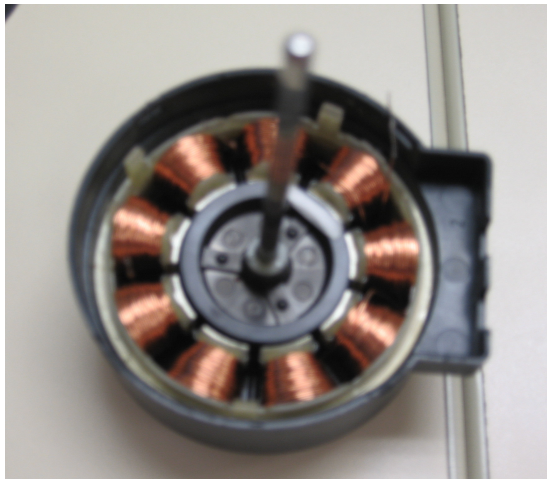


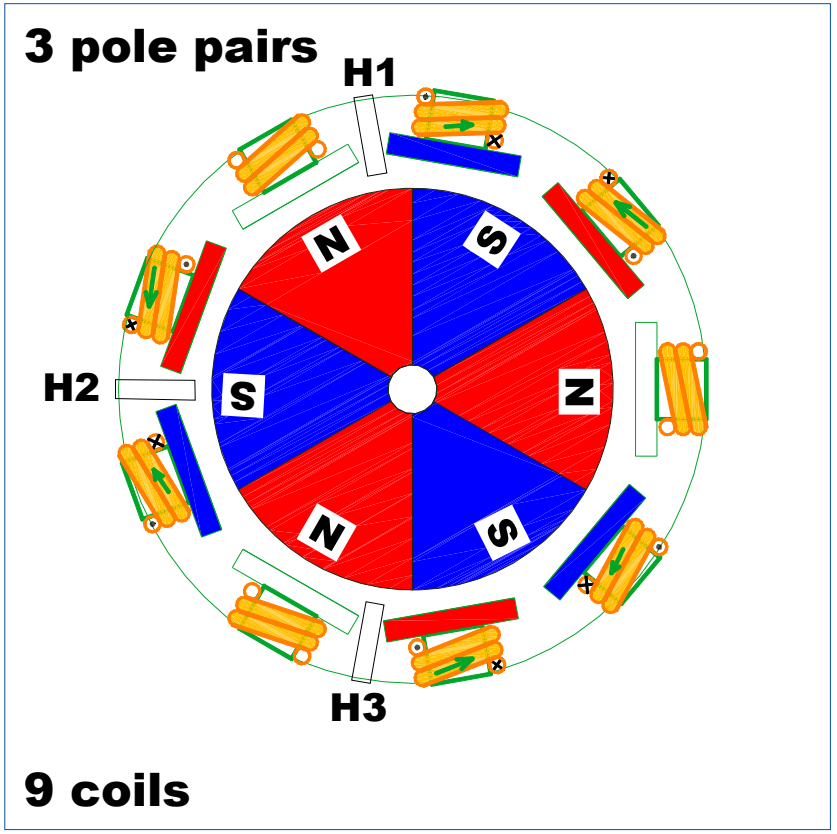




freescale™ semiconductor

BLDC motor configuration

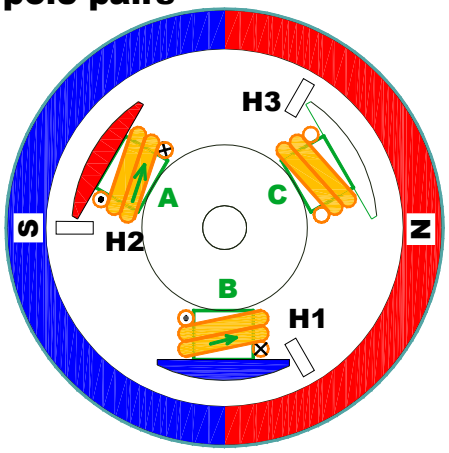




freescale™ semiconductor

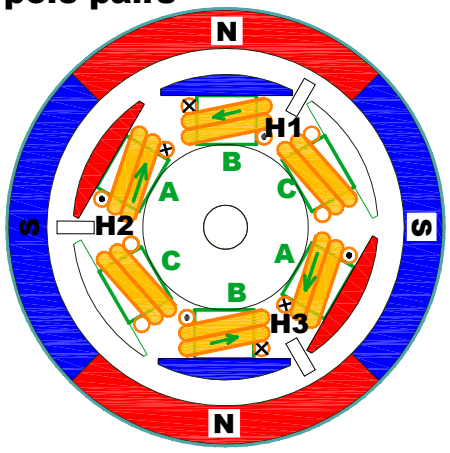
External Rotor – Different Motor Configurations

1 pole pairs



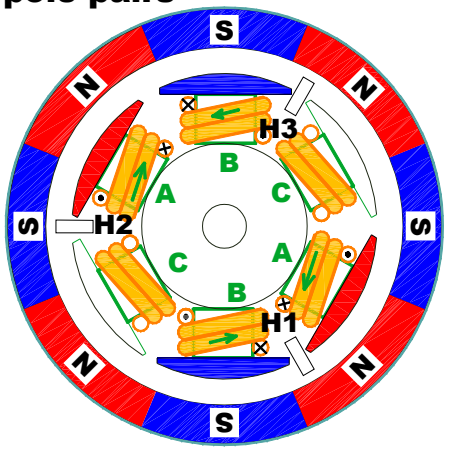
H1	H2	H3
1	0	1

2 pole pairs



H1	H2	H3
1	0	1

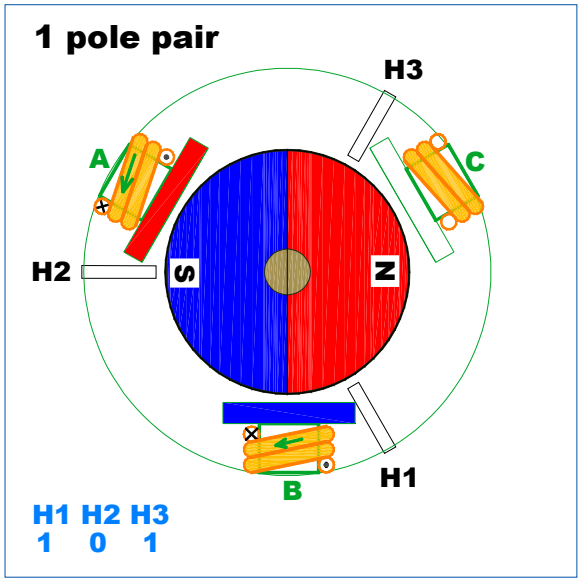
4 pole pairs

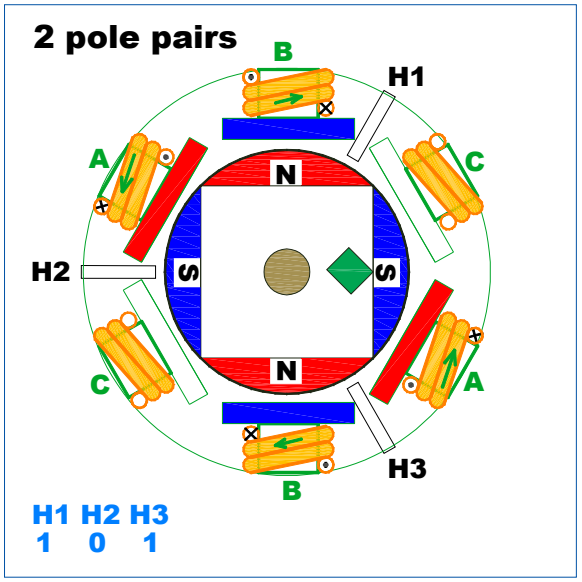


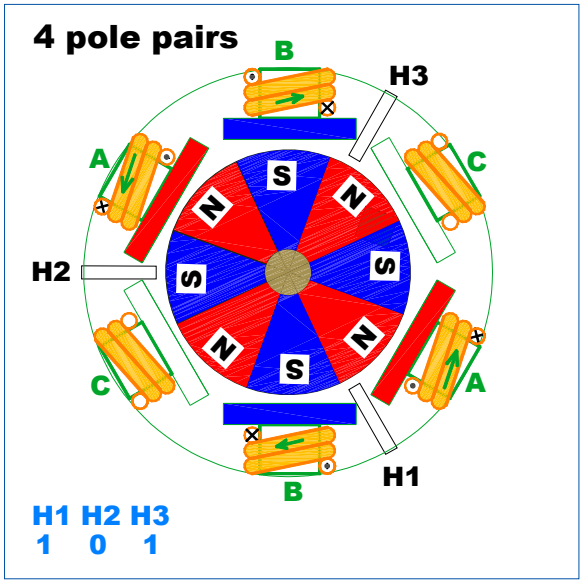
H1	H2	H3
1	0	1

freescale™ semiconductor

Internal Rotor – Different Motor Configurations



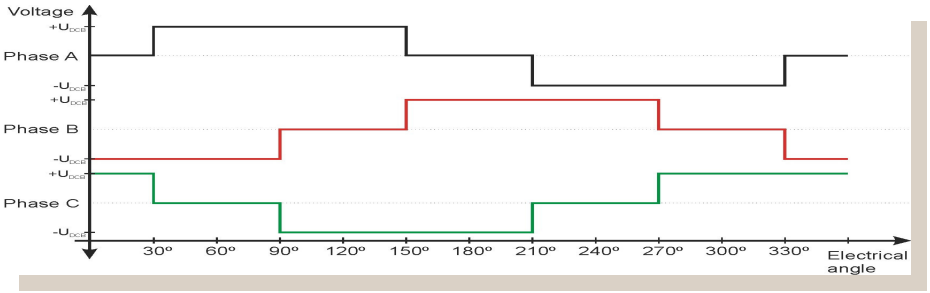




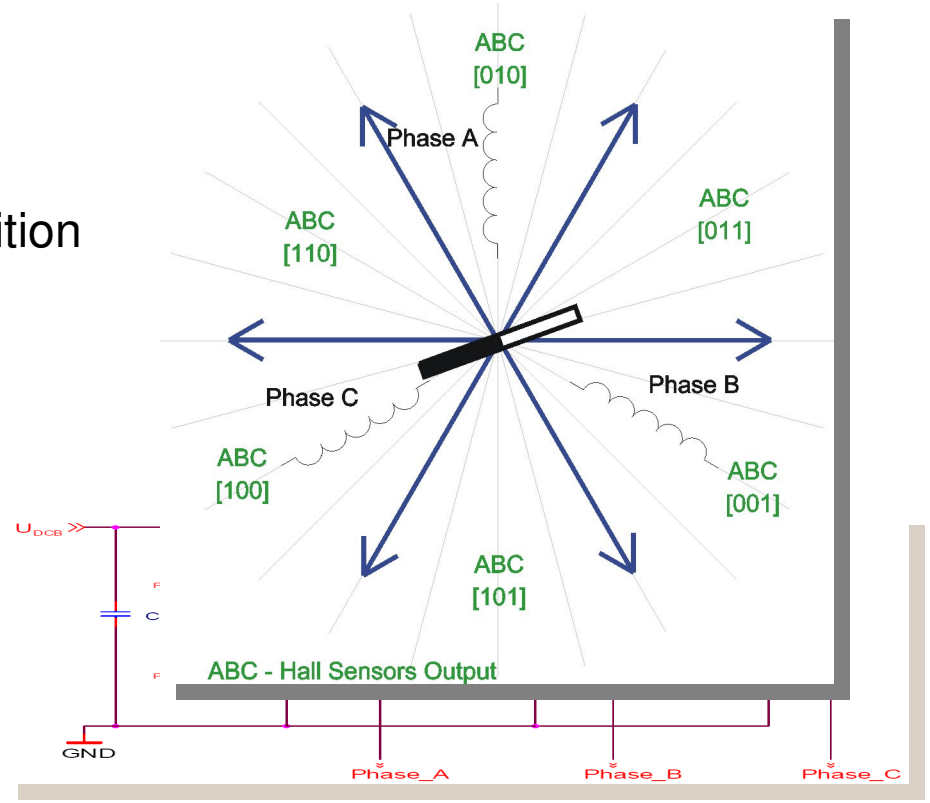
freescale™ semiconductor

Six Step BLDC Motor Control

- Voltage applied on two phases only
- It creates 6 flux vectors
- Phases are power based on rotor position
- The process is called commutation



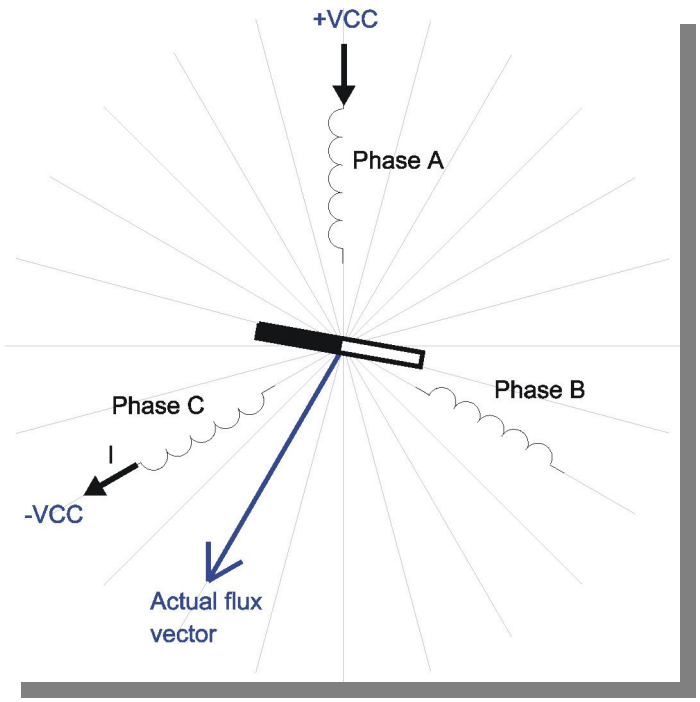
Phases voltage



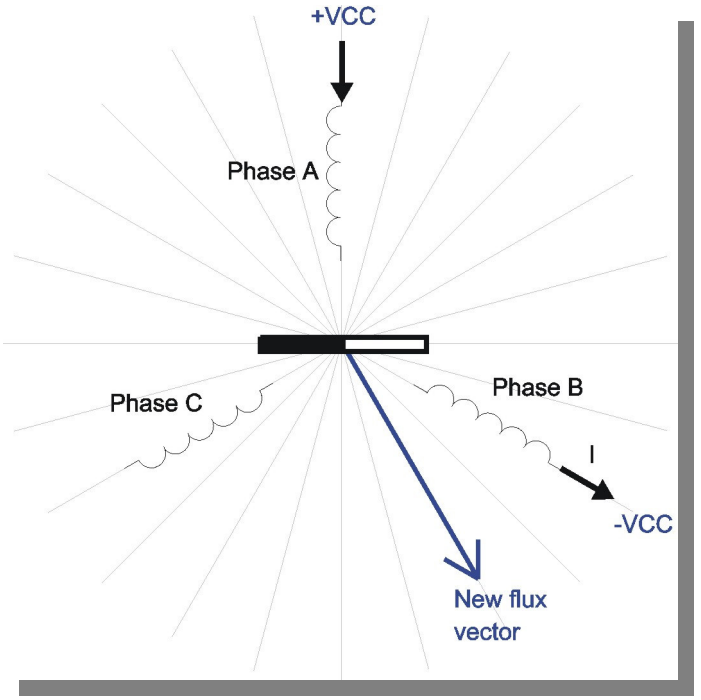
Power Stage

freescale semiconductor

- Commutation example
  - Stator field is maintained 60°, 120° relative to rotor field



Before commutation

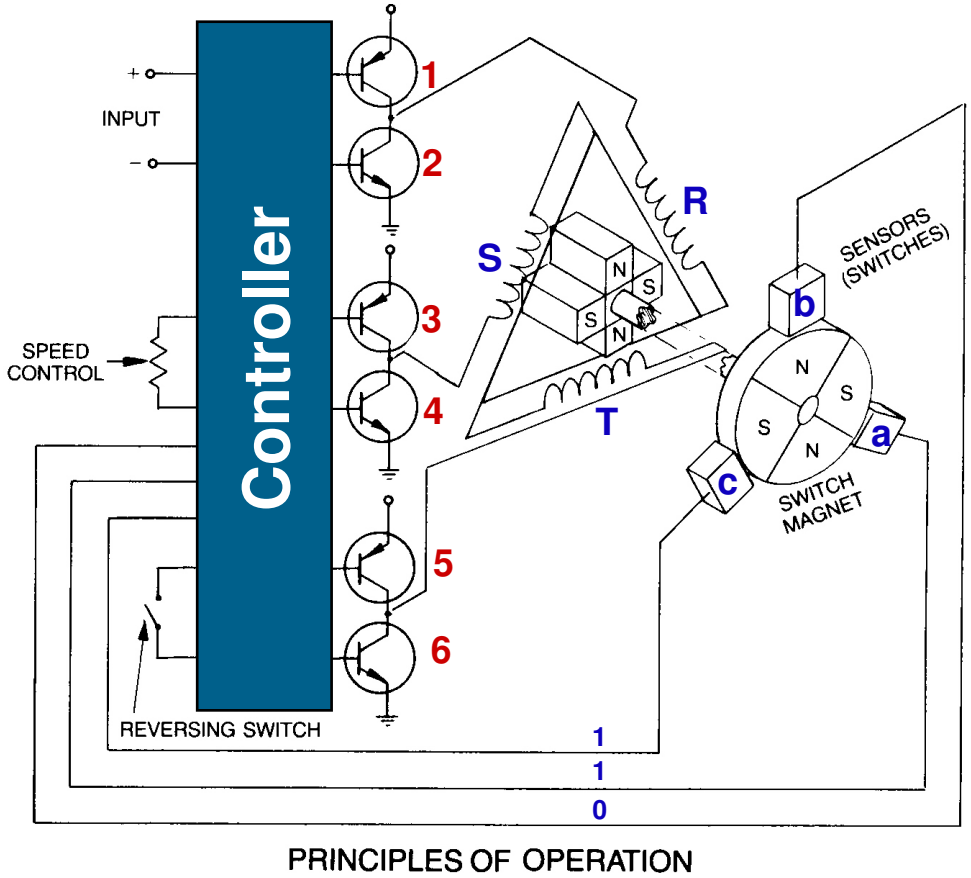


After commutation

freescale™ semiconductor

Brushless DC Motor Control

- Six Step BLDC Motor Control cont'd



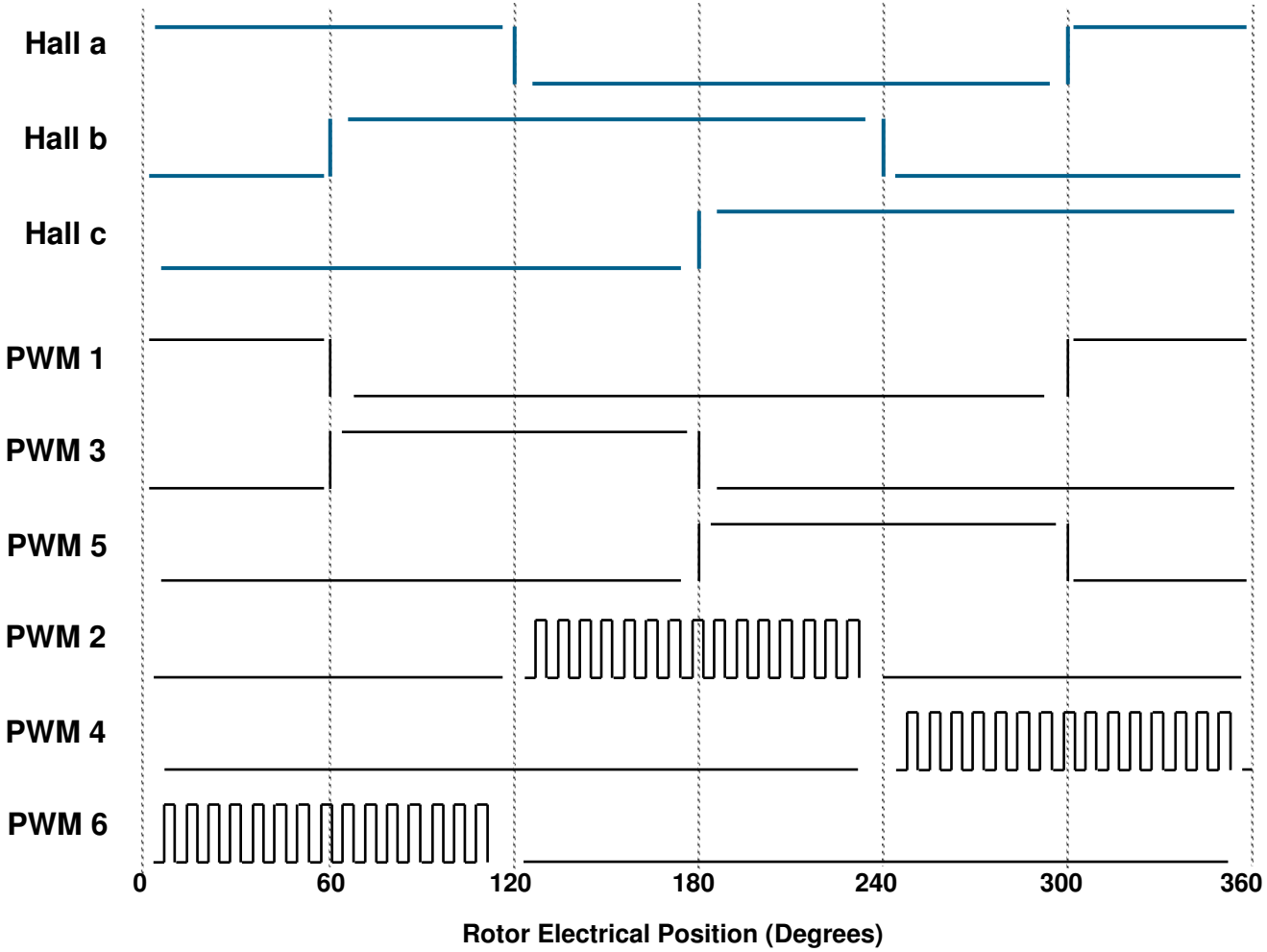
PRINCIPLES OF OPERATION

Source: Eastern Air Devices, Inc. Brushless DC Motor Brochure

freescale semiconductor

Brushless DC Motor Control

- Six Step BLDC Motor Control cont'd



freescale™ semiconductor

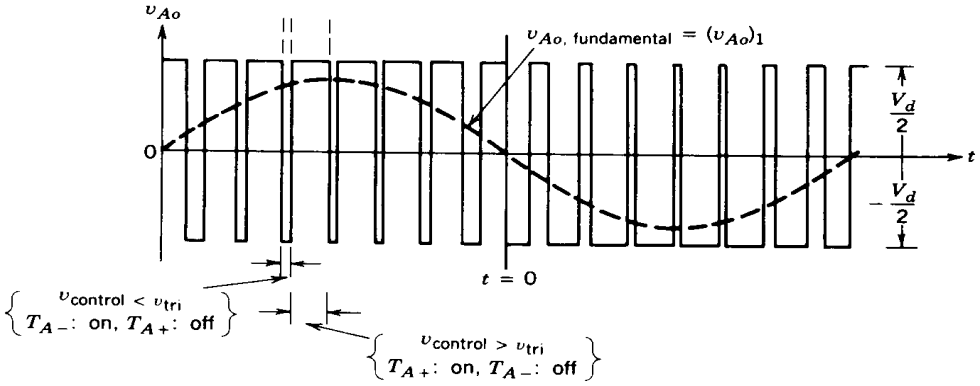
Brushless DC Motor Control

- Example of commutation table

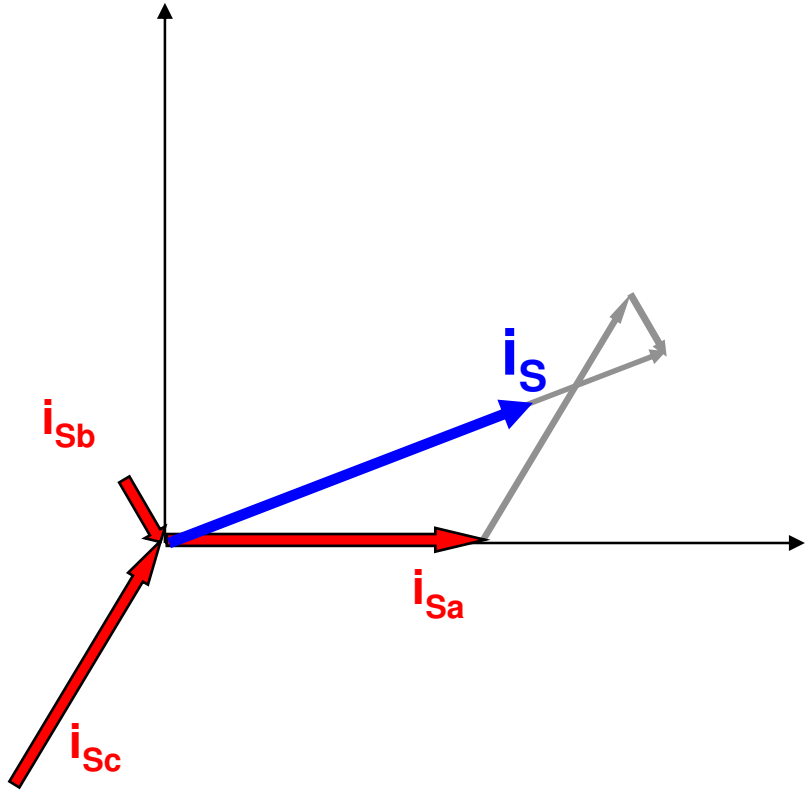
Hall Sensor A	Hall Sensor B	Hall Sensor C	Phase A	Phase B	Phase C
1	0	0	$-V_{DCB}$	$+V_{DCB}$	NC
1	0	1	NC	$+V_{DCB}$	$-V_{DCB}$
0	0	1	$+V_{DCB}$	NC	$-V_{DCB}$
0	1	1	$+V_{DCB}$	$-V_{DCB}$	NC
0	1	0	NC	$-V_{DCB}$	$+V_{DCB}$
1	1	0	$-V_{DCB}$	NC	$+V_{DCB}$

Brushless DC Motor Control

- Sinusoidal BLDC motor control



All three phases are powered by sinewave shifted by 120°



We are able to generate stator field to any position over 360°

freescale™ semiconductor

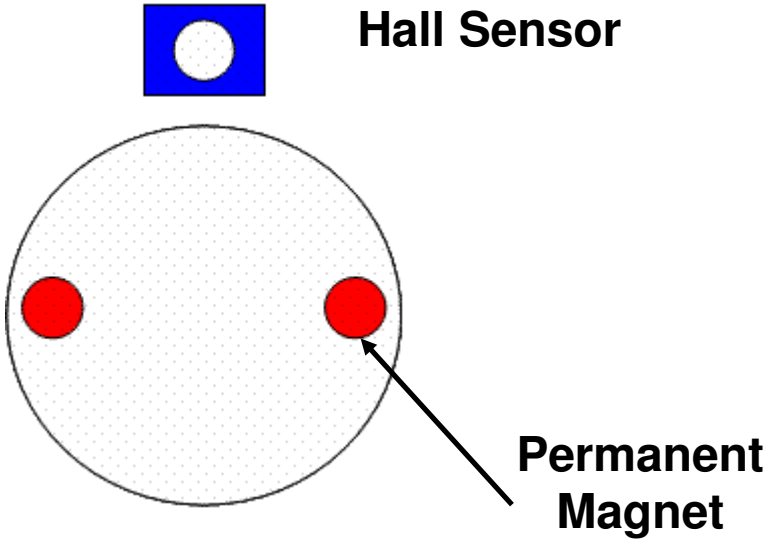
Brushless DC Motor Control Summary

- Six step control versus sinusoidal control

Six step control	Sinusoidal control
+ Simple PWM generation	– More complex PWM generation (sinewave has to be generated)
– Ripple in the torque (stator flux jumps by 60°)	+ Smooth torque (stator flux rotates fluently)
– A little noise operation (due to ripple in the torque)	+ Very quiet
+ Simple sensor	– Requires sensor with high resolution

Sensor Example: Hall Effect Sensor

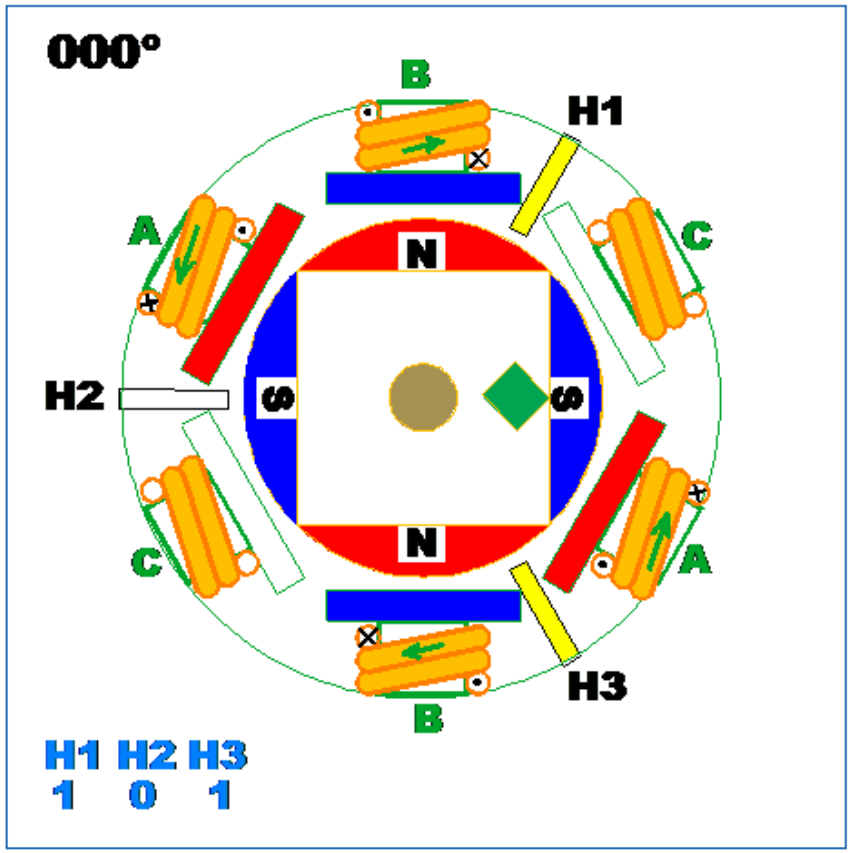
- Hall effect sensor is a transducer that varies its output voltage in response to changes in magnetic field
- Hall sensors are used for proximity switching, positioning, speed detection and current sensing applications
- In this case, hall sensors are used in On/Off mode



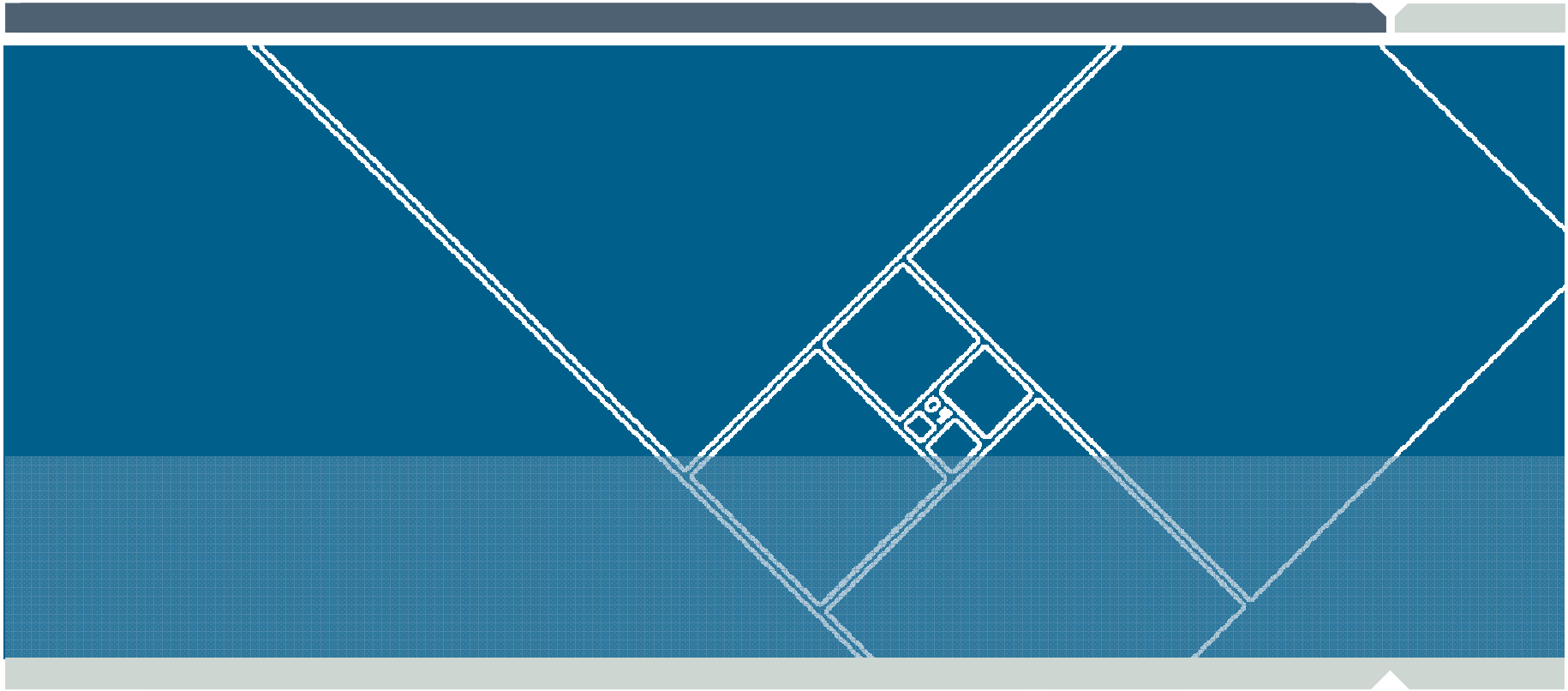
Everytime a magnetic field is sensed, a change in voltage can be detected

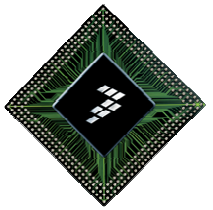
freescale™ semiconductor

Putting All Together



freescale™ semiconductor

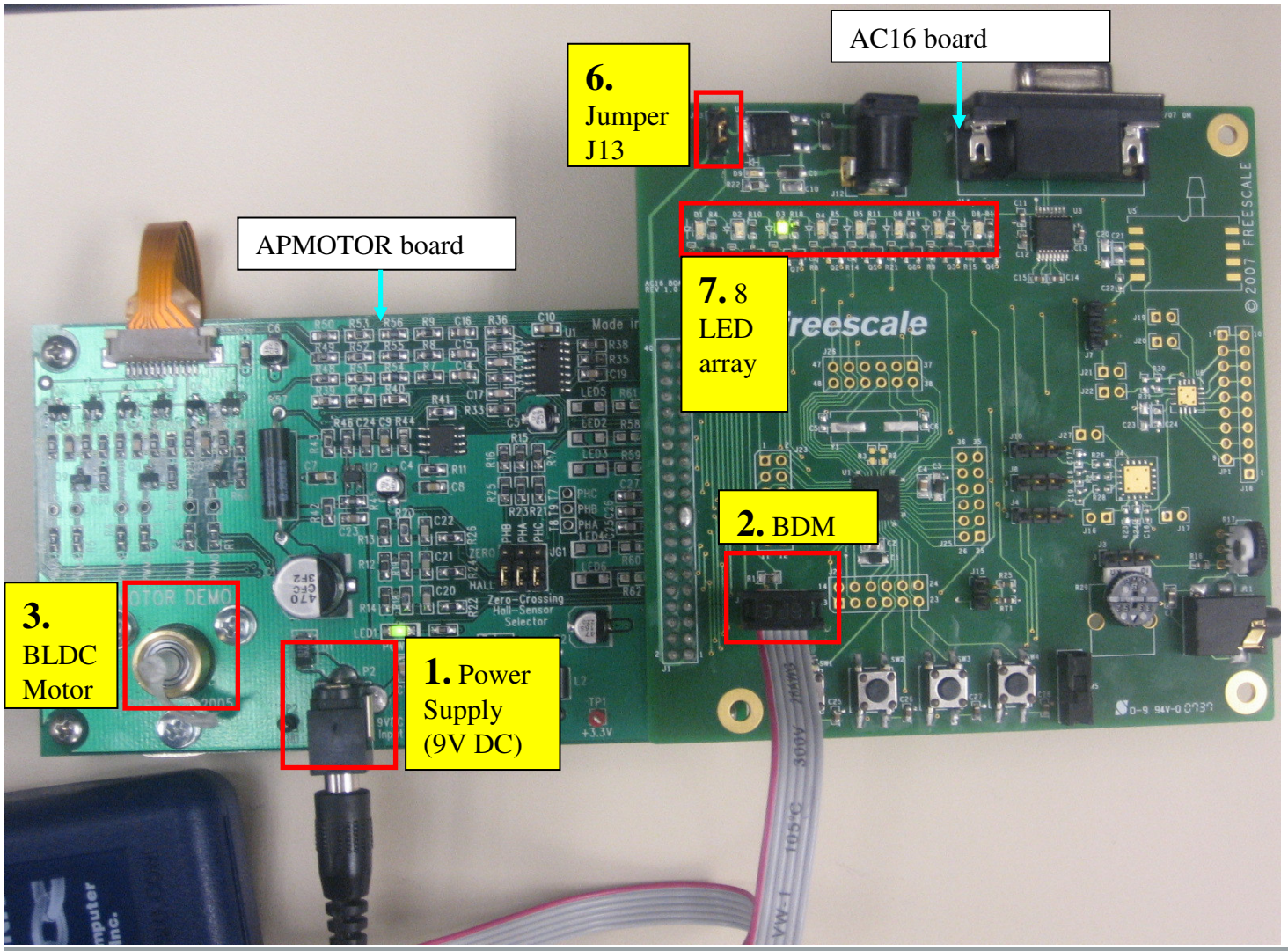


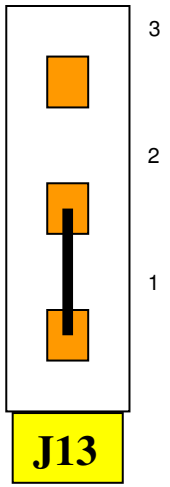


Lab1

freescale™
semiconductor

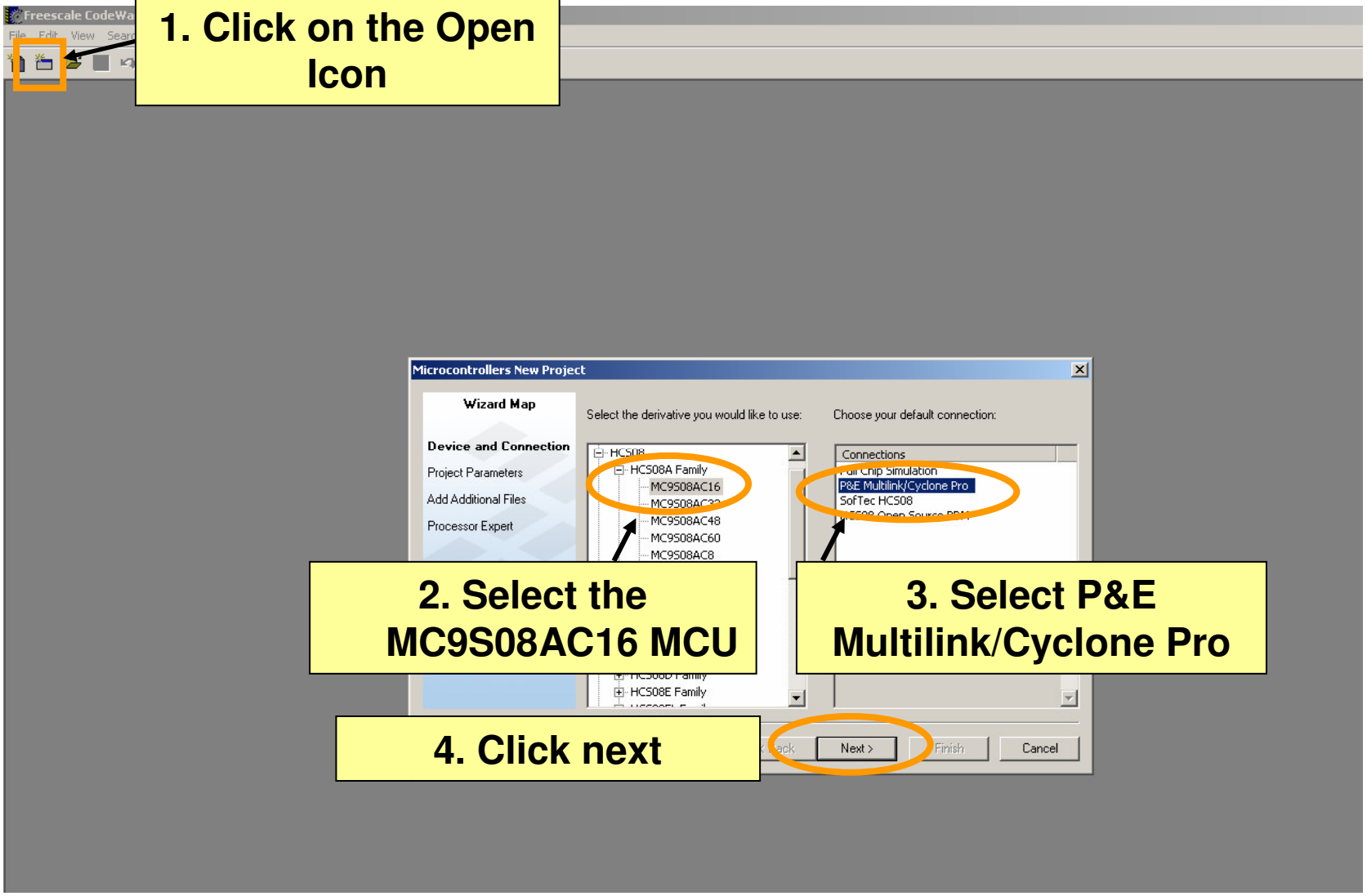
How to Set Up the Boards





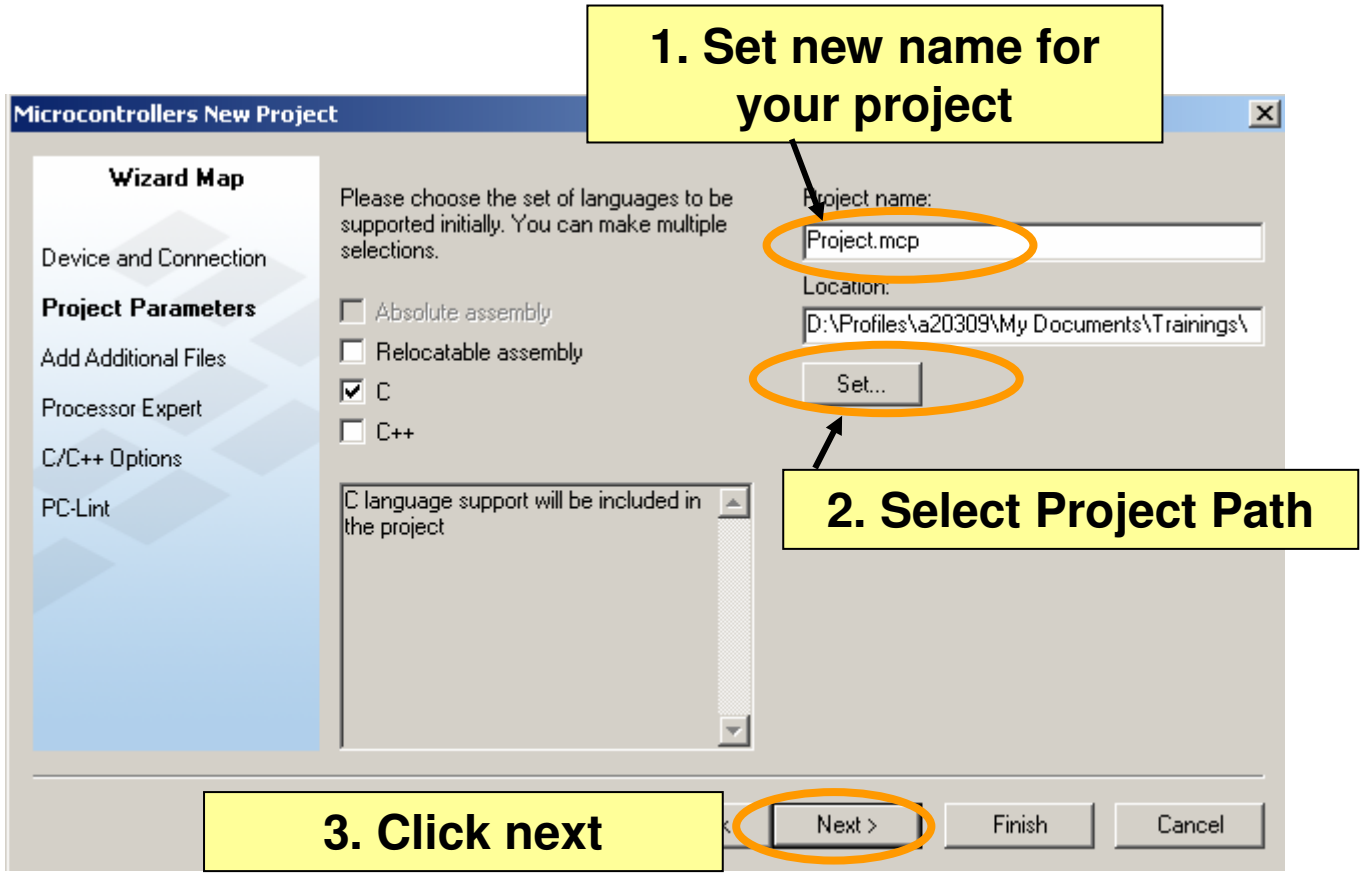
freescale semiconductor

Create a New CodeWarrior Project



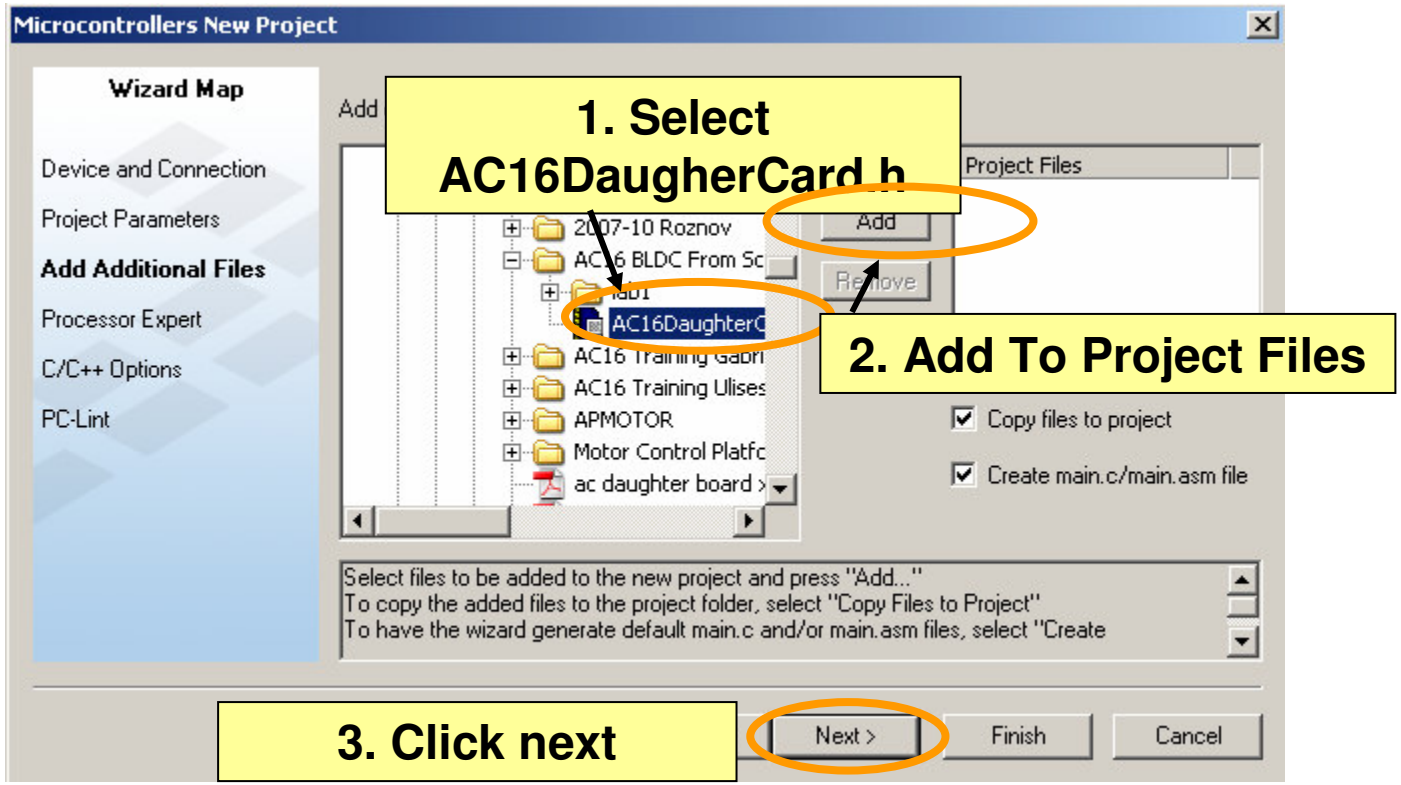
freescale semiconductor

Name Your CodeWarrior Project



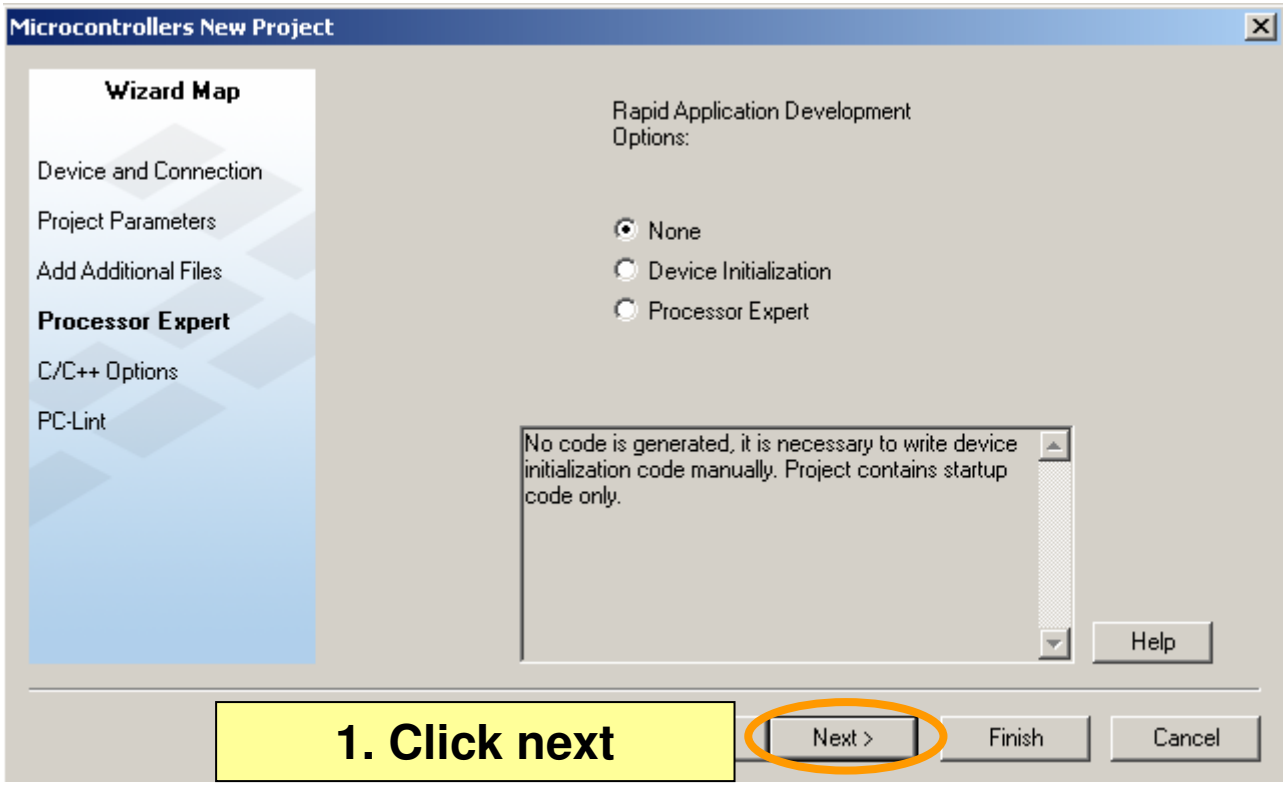
freescale™ semiconductor

Add Additional Files



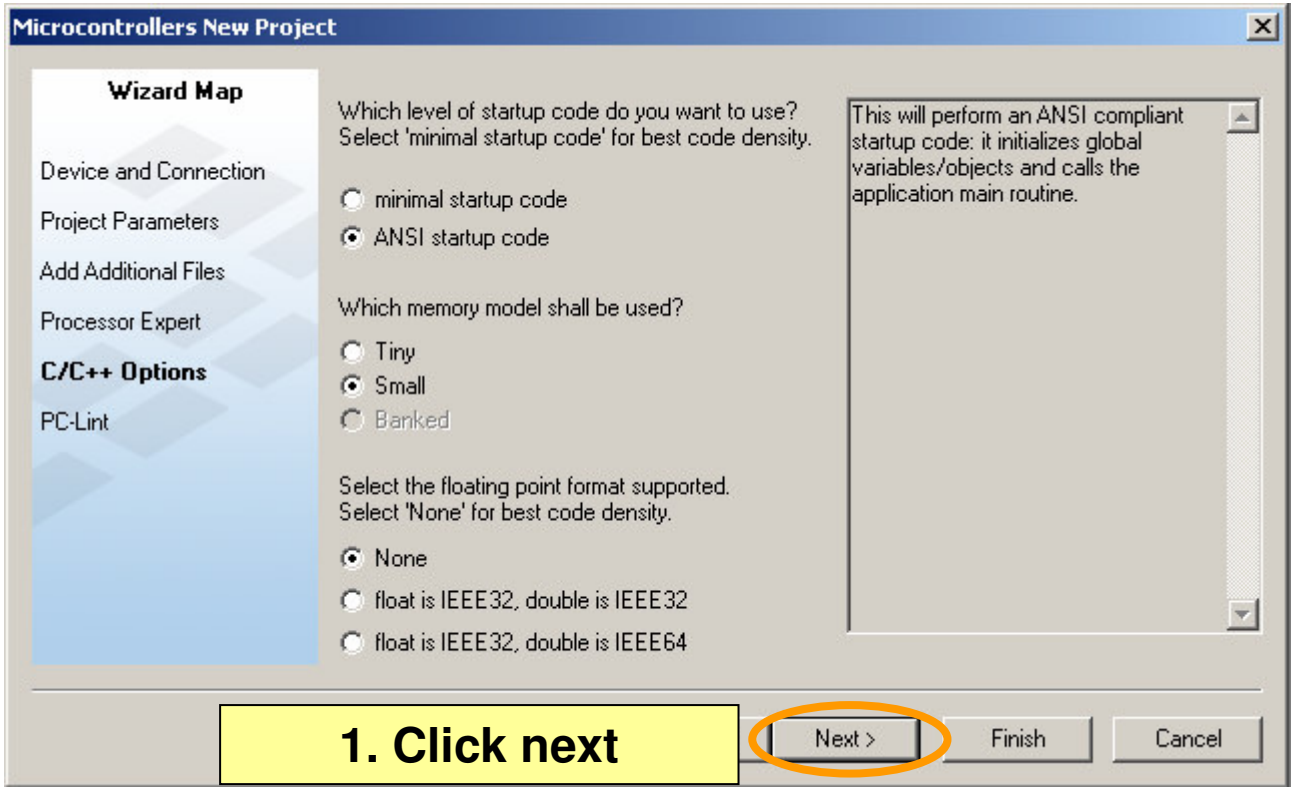
freescale semiconductor

Processor Expert



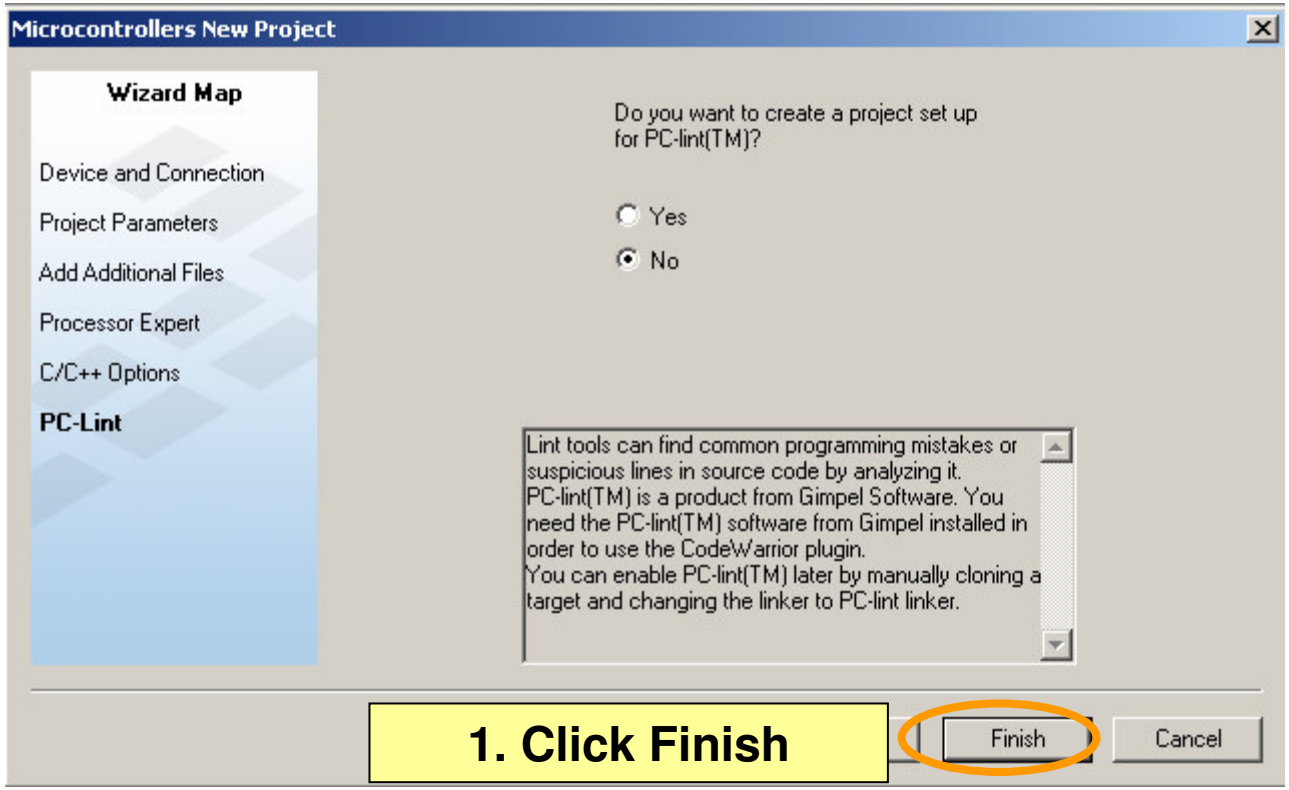
freescale semiconductor

C Options



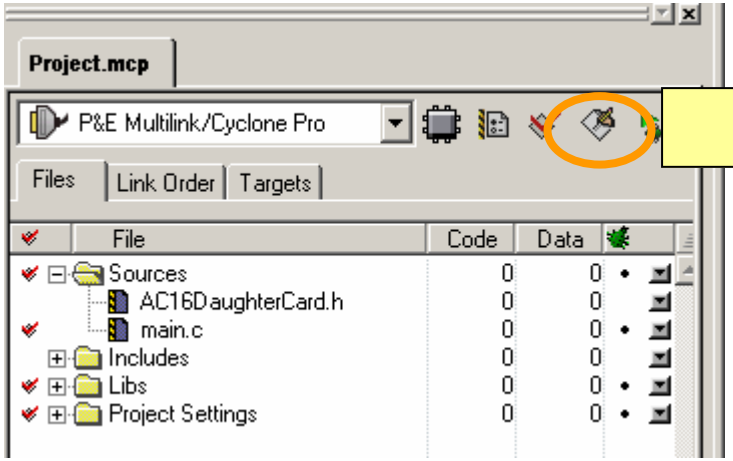
freescale™ semiconductor

PC-Lint Options



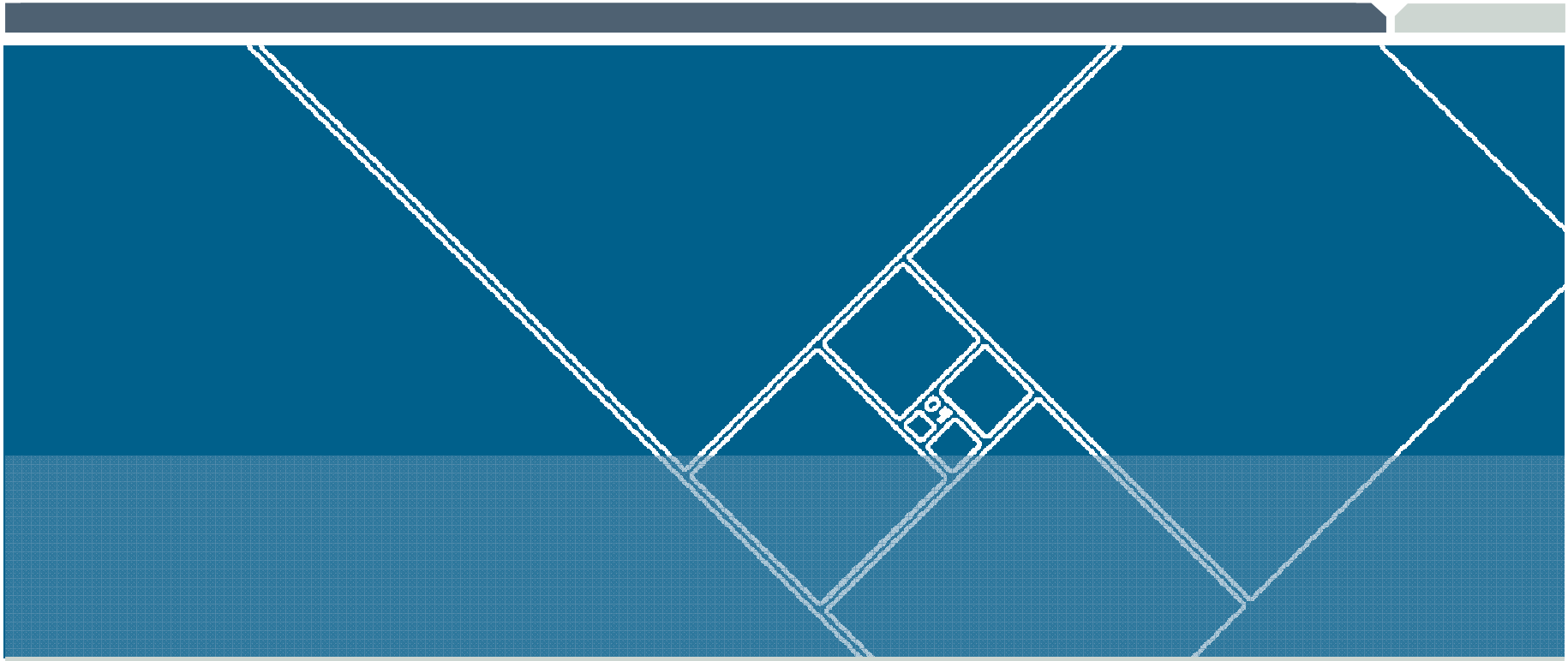
freescale™ semiconductor

New Project Set Up

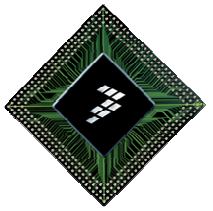


1. Click On Make

freescale™ semiconductor



Commutation Table & Knowing Position with Hall Effect sensors



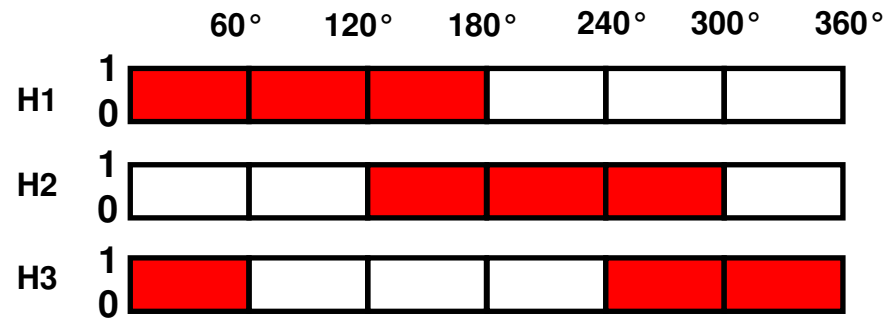
freescale™ semiconductor

Necessity of Knowing the Position

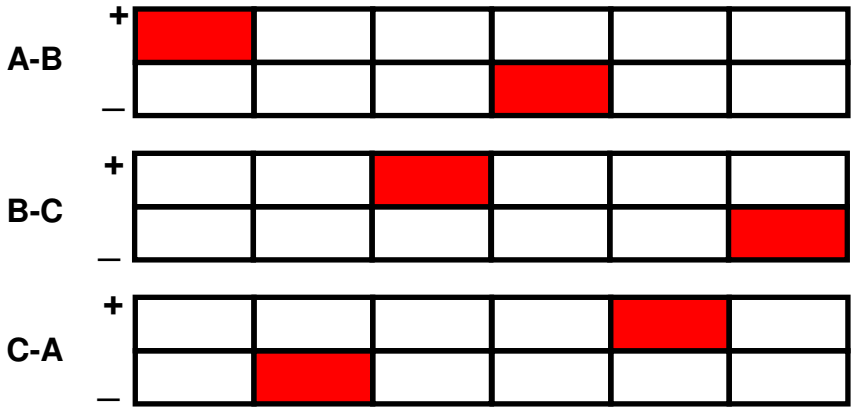
- To spin 3-phase BLDC motor:
  - Detect position/commutation
  - Read commutation table
  - Mask and swap phases

It is important to know the rotor position in order to maintain the rotating magnetic field

Signal sequence diagram for the hall sensors

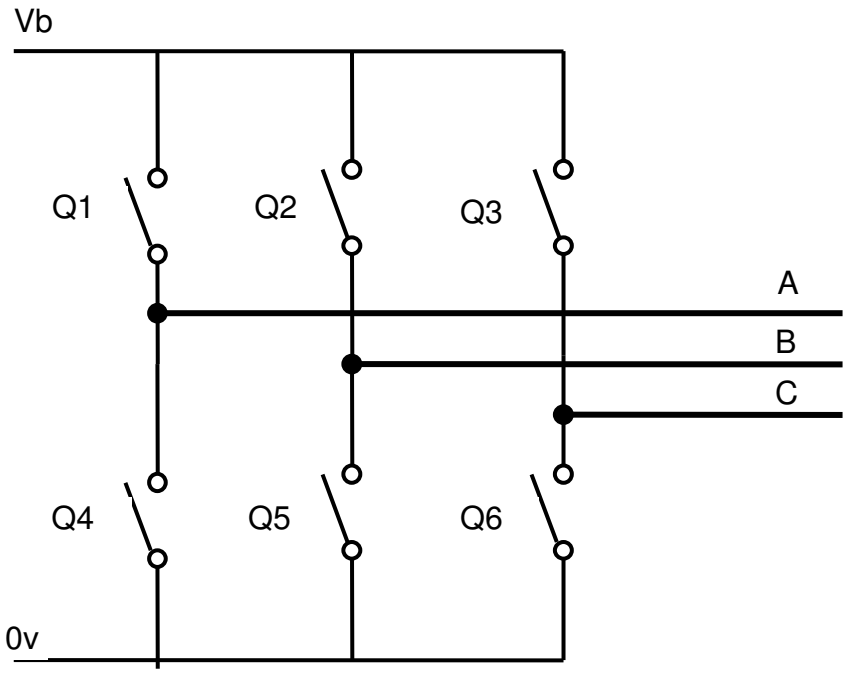


Supplied motor voltage



freescale™ semiconductor

3-Phase Inverter

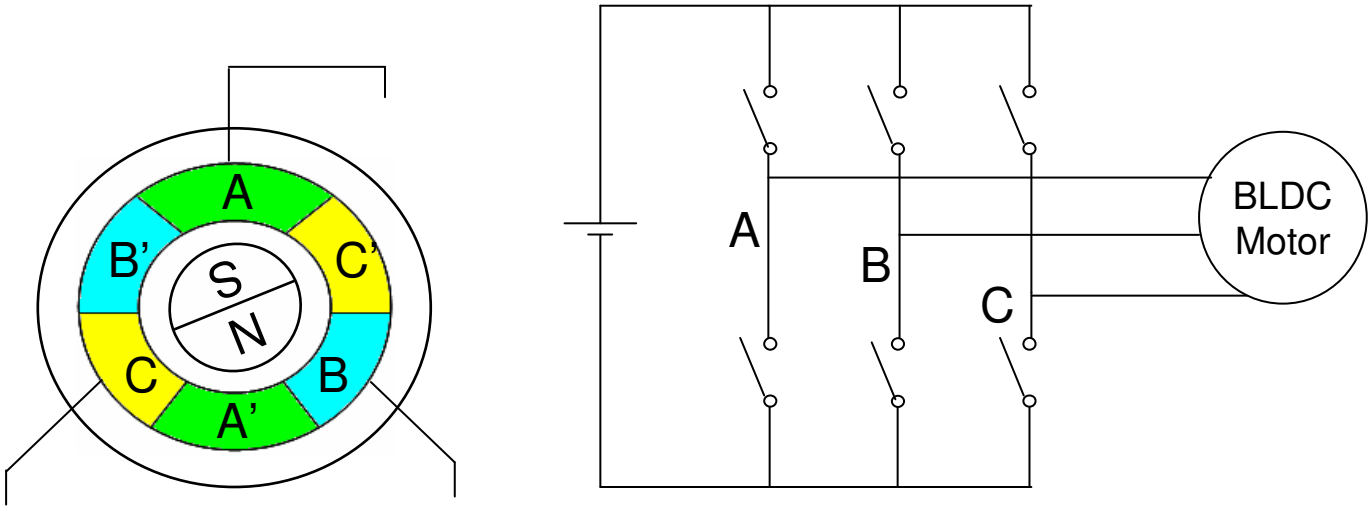


With the 3-phase inverter, you can control which phases need to be fed in order to turn the motor

Q1, Q2 and Q3 is where the current goes in the motor and Q4, Q5 and Q6 is where the current goes out of the motor

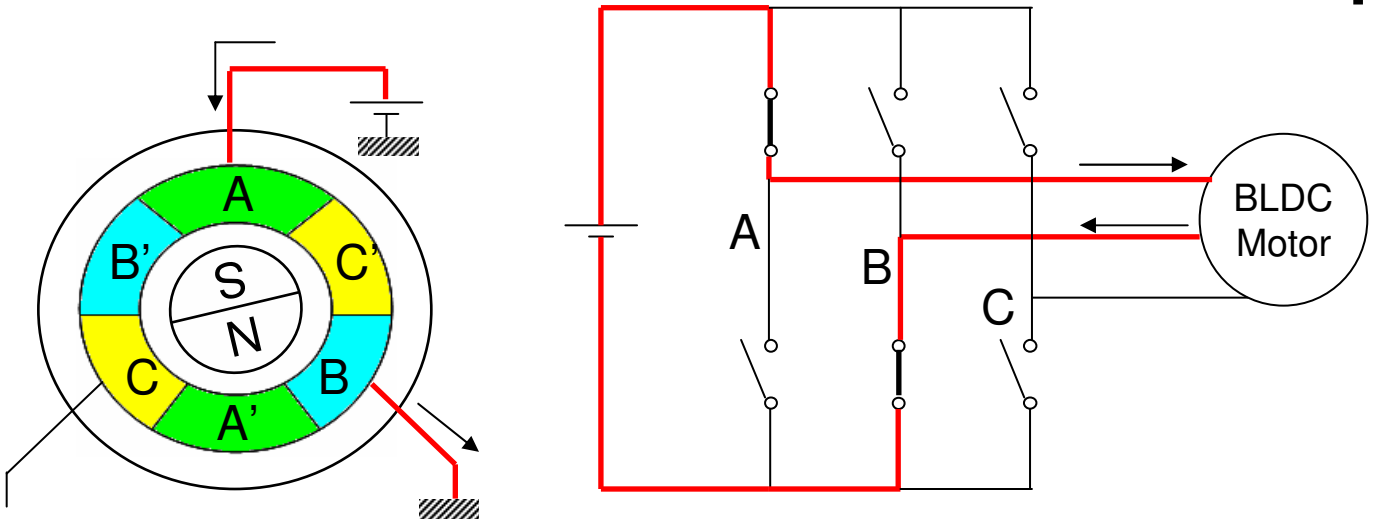
freescale™ semiconductor

Control of 3-Phase Inverter Determined on the Hall Sensor Position



freescale™ semiconductor

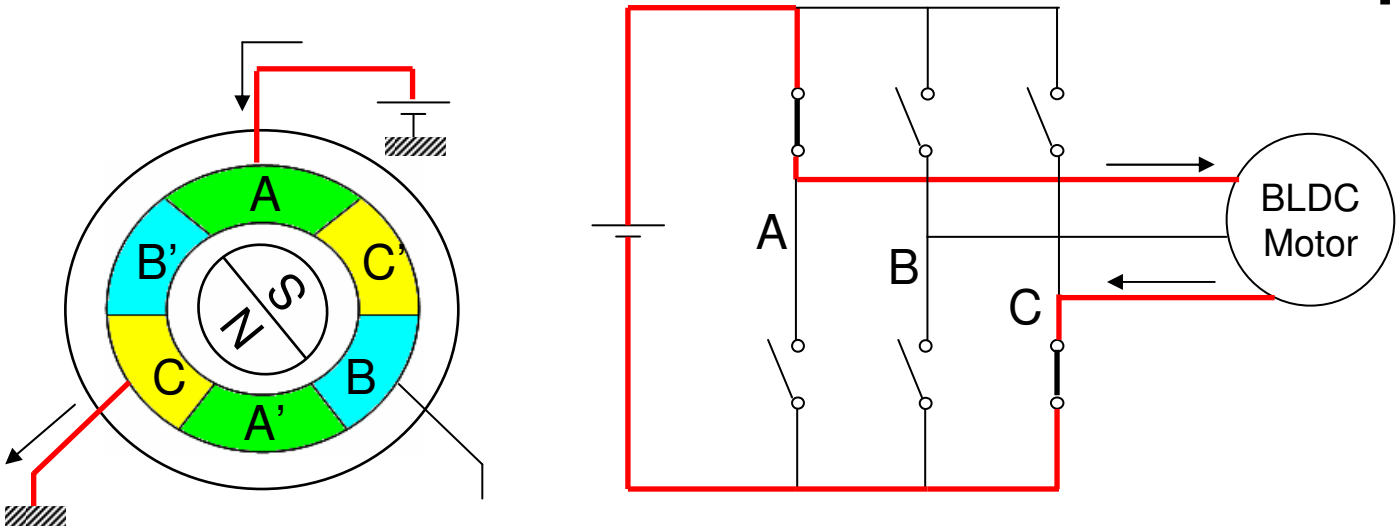
Control of 3-Phase Inverter Determined on the Hall Sensor Position



H1	H2	H3	Phase A	Phase B	Phase C
1	0	1	$+V_{DCB}$	$-V_{DCB}$	NC
1	0	0	$+V_{DCB}$	NC	$-V_{DCB}$
1	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	0	$-V_{DCB}$	$+V_{DCB}$	NC
0	1	1	$-V_{DCB}$	NC	$+V_{DCB}$
0	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

freescale™ semiconductor

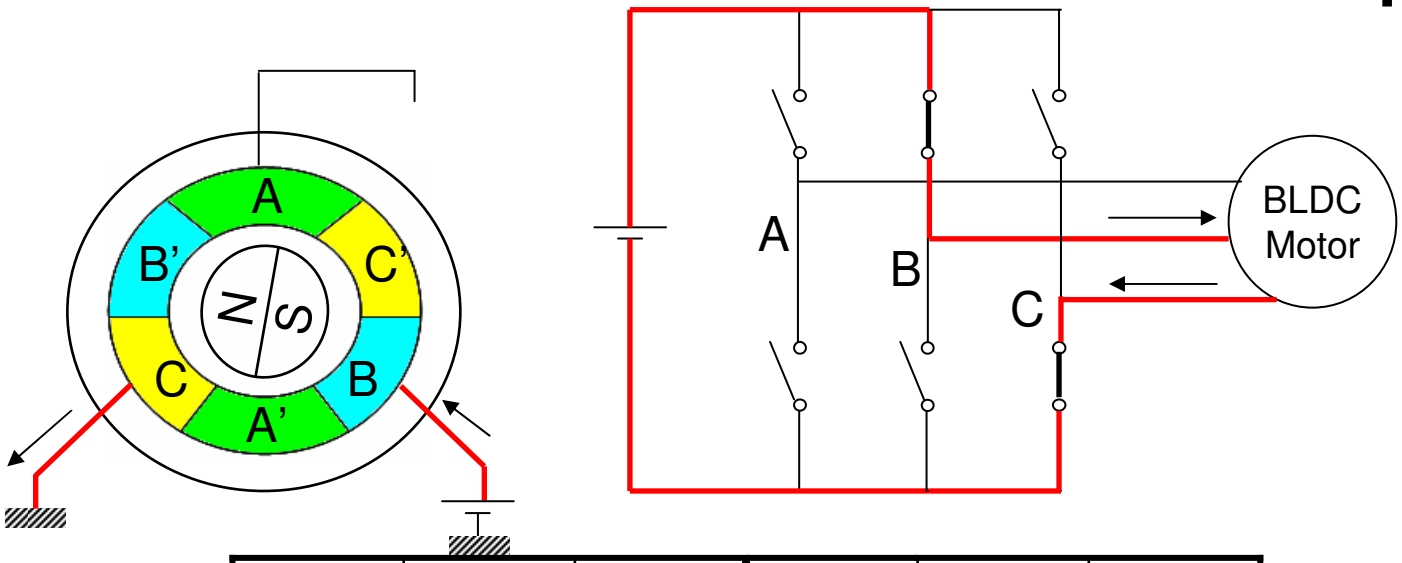
Control of 3-Phase Inverter Determined on the Hall Sensor Position



H1	H2	H3	Phase A	Phase B	Phase C
1	0	1	$+V_{DCB}$	$-V_{DCB}$	NC
1	0	0	$+V_{DCB}$	NC	$-V_{DCB}$
1	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	0	$-V_{DCB}$	$+V_{DCB}$	NC
0	1	1	$-V_{DCB}$	NC	$+V_{DCB}$
0	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

freescale™ semiconductor

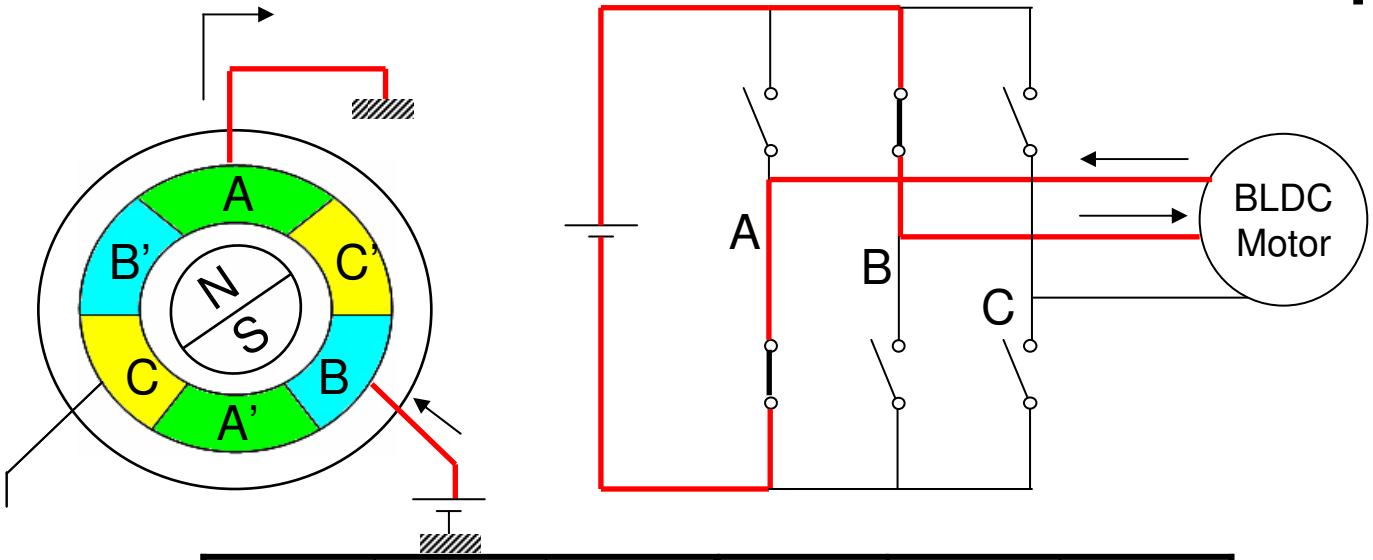
Control of 3-Phase Inverter Determined on the Hall Sensor Position



H1	H2	H3	Phase A	Phase B	Phase C
1	0	1	$+V_{DCB}$	$-V_{DCB}$	NC
1	0	0	$+V_{DCB}$	NC	$-V_{DCB}$
1	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	0	$-V_{DCB}$	$+V_{DCB}$	NC
0	1	1	$-V_{DCB}$	NC	$+V_{DCB}$
0	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

freescale™ semiconductor

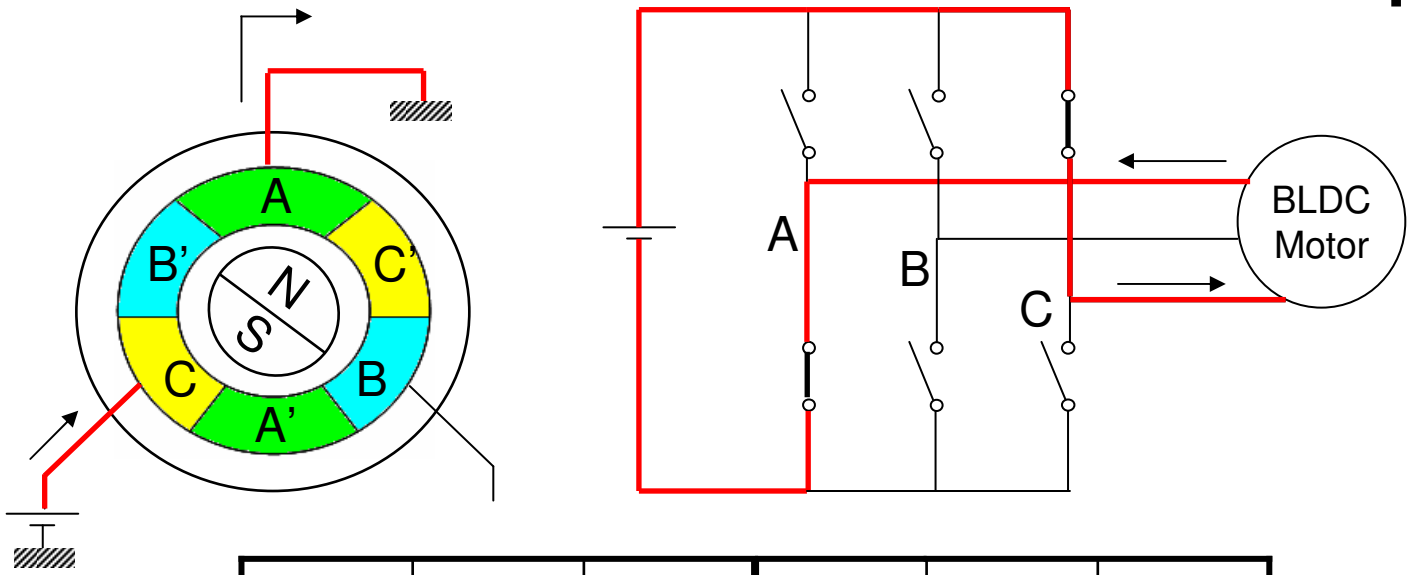
Control of 3-Phase Inverter Determined on the Hall Sensor Position



H1	H2	H3	Phase A	Phase B	Phase C
1	0	1	$+V_{DCB}$	$-V_{DCB}$	NC
1	0	0	$+V_{DCB}$	NC	$-V_{DCB}$
1	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	0	$-V_{DCB}$	$+V_{DCB}$	NC
0	1	1	$-V_{DCB}$	NC	$+V_{DCB}$
0	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

freescale™ semiconductor

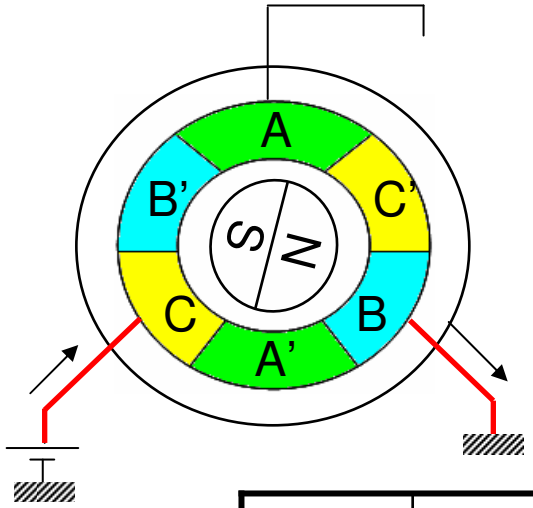
Control of 3-Phase Inverter Determined on the Hall Sensor Position

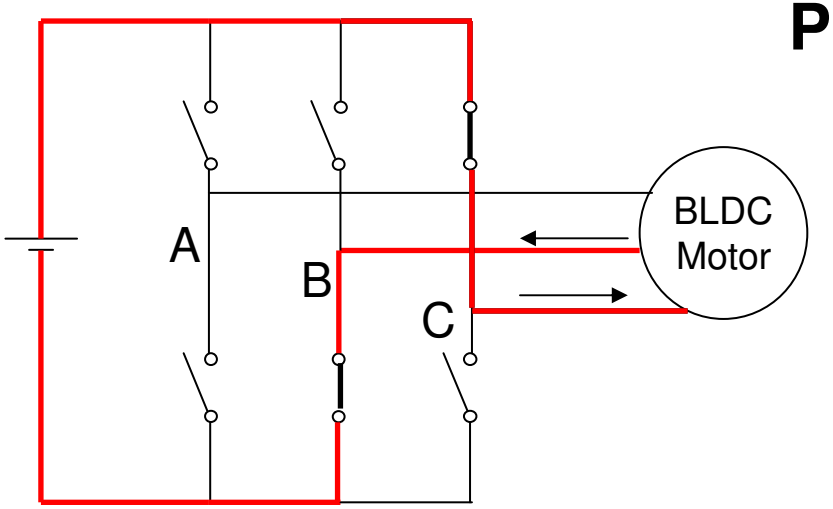


H1	H2	H3	Phase A	Phase B	Phase C
1	0	1	$+V_{DCB}$	$-V_{DCB}$	NC
1	0	0	$+V_{DCB}$	NC	$-V_{DCB}$
1	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	0	$-V_{DCB}$	$+V_{DCB}$	NC
0	1	1	$-V_{DCB}$	NC	$+V_{DCB}$
0	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

freescale™ semiconductor

Control of 3-Phase Inverter Determined on the Hall Sensor Position





H1	H2	H3	Phase A	Phase B	Phase C
1	0	1	$+V_{DCB}$	$-V_{DCB}$	NC
1	0	0	$+V_{DCB}$	NC	$-V_{DCB}$
1	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	0	$-V_{DCB}$	$+V_{DCB}$	NC
0	1	1	$-V_{DCB}$	NC	$+V_{DCB}$
0	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

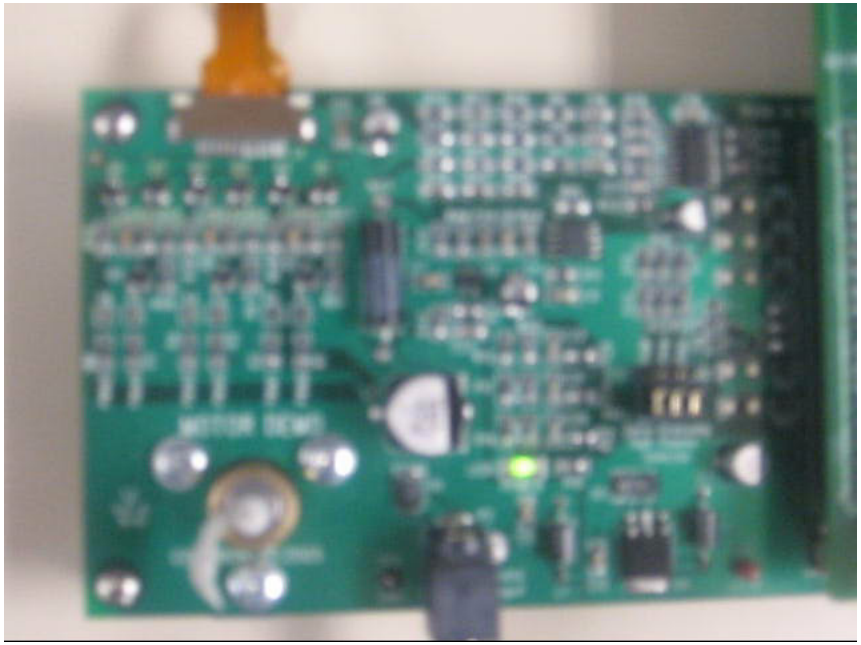
freescale™ semiconductor

Lab 1

- Make a program that moves the motor in the clockwise direction
  - On/Off transistors

TO DO:

- Enable Switch1 to enable commutations
- Enable 3-Phase Inverter
- LEDs will still reflect HALL Effect sensors
- Each time Switch1 is pressed, we will advance one step in the commutation table
- Commutation table will tell us which transistors to turn on
- When Switch1 is not pressed, turn OFF transistors



freescale™ semiconductor

Import Table Add Variables, Enable Switch1, Enable Inverter

```
extern unsigned char table_rotate[8];
unsigned char value;
unsigned char commutation;
void main(void) {

  EnableInterrupts; /* enable interrupts */
  /* include your code here */
  ENABLESWITCH(1);
  ENABLELED(1);
  ENABLELED(2);
  ENABLELED(3);
  ENABLE3PHASEINVERTER();
  for(;;) {
```

Wait for Switch1 to be Pressed

```
for(;;) {
  __RESET_WATCHDOG(); /* feeds the dog */
  LED1_PIN = HALL_1;
  LED2_PIN = HALL_2;
  LED3_PIN = HALL_3;
  if(!SW1_PIN)
  {

  }
```

Once Pressed, Advance Counter, Commutate

```
for(;;) {
  __RESET_WATCHDOG(); /* feeds the dog */
  LED1_PIN = HALL_1;
  LED2_PIN = HALL_2;
  LED3_PIN = HALL_3;
  if(!SW1_PIN)
  {
      value++;
      if(value>=7)  value = 1;
      commutation = table_rotate[value];
      if(commutation & Q1_MASK) Q1 = 1;
      if(commutation & Q4_MASK) Q4 = 1;
      if(commutation & Q2_MASK) Q2 = 1;
      if(commutation & Q5_MASK) Q5 = 1;
      if(commutation & Q3_MASK) Q3 = 1;
      if(commutation & Q6_MASK) Q6 = 1;
}
```

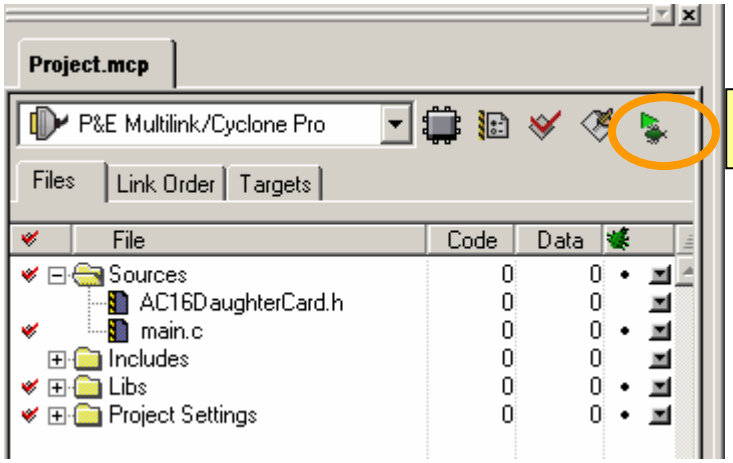
Wait for Switch to be Released

```
  if(commutation & Q3_MASK) Q3 = 1;
  if(commutation & Q6_MASK) Q6 = 1;
  while(!SW1_PIN)
  {
    __RESET_WATCHDOG();
    LED1_PIN = HALL_1;
    LED2_PIN = HALL_2;
    LED3_PIN = HALL_3;
} }
```

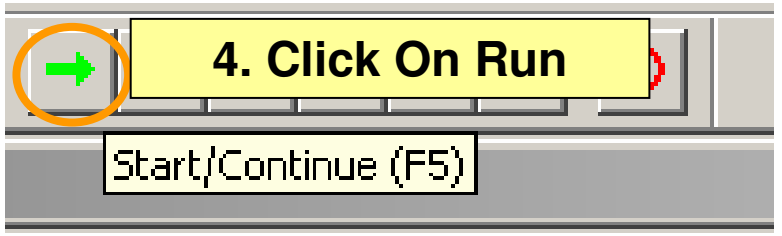
```
TURNOFFTRANSISTORS();
```

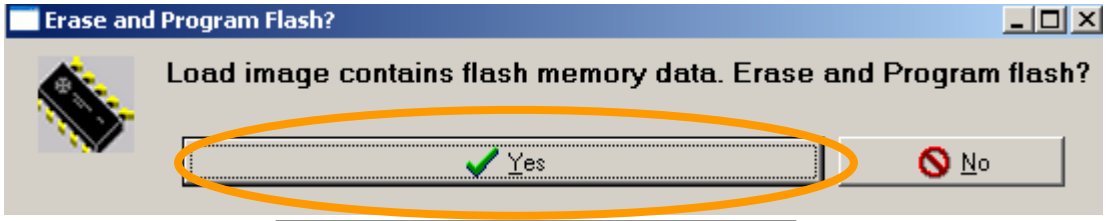
IT IS IMPORTANT TO TURN OFF TRANSISTORS!

Download and Run Code

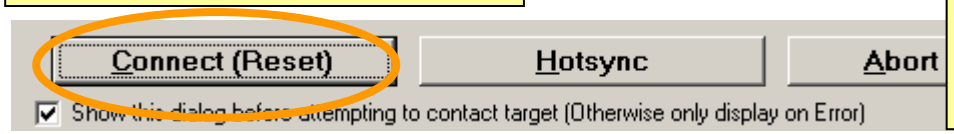


1. Click On Run



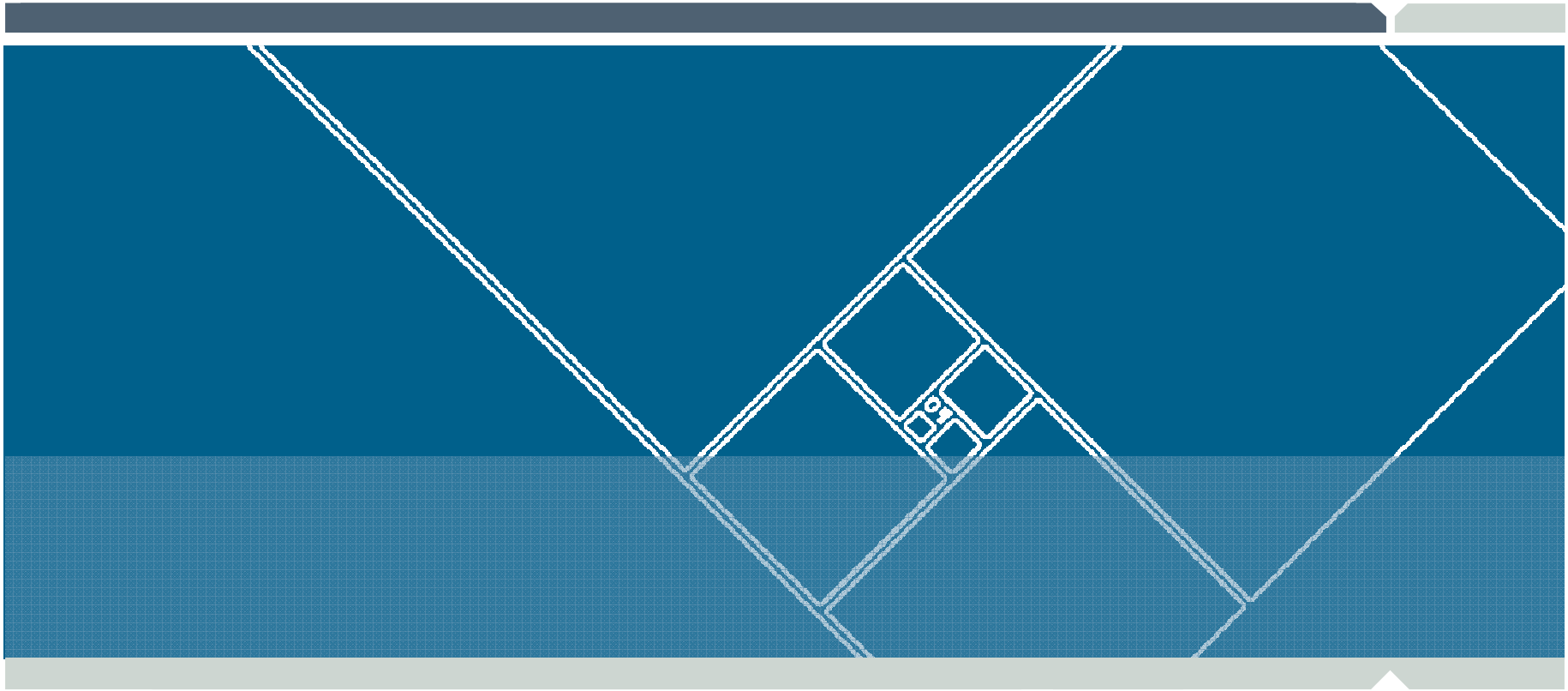


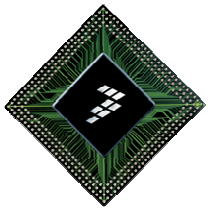
2. Click On Connect on the Debugger



3. Click On Yes To Reprogram the MCU

freescale semiconductor





Lab2

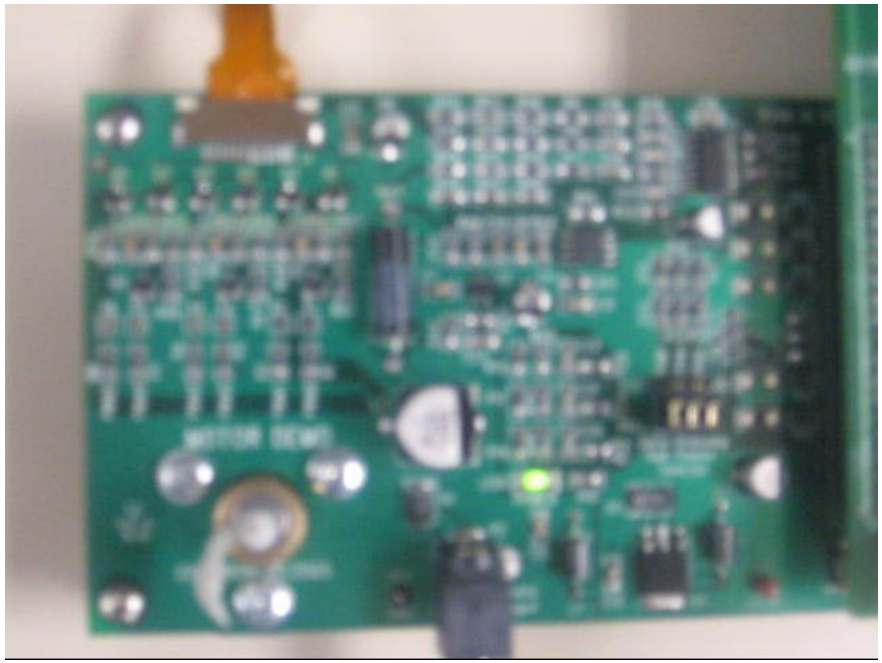
freescale™
semiconductor

Lab 2

- Make a program that move the motor in clockwise direction
  - On/Off transistors

TO DO:

- Enable Switch5 to enable commutations
- Check Hall Effect sensors to evaluate which commutation state motor is in
- When Hall Effect sensors changes state, transistors will commutate



freescale ™
semiconductor

Add Variables, Enable Switch5

```
unsigned char pasthallsensors;
unsigned char hallsensors;
 void main(void) {

   EnableInterrupts; /* enable interrupts */
   /* include your code here */
   ENABLESWITCH(5);
   ENABLELED(1);
   ENABLELED(2);
   ENABLELED(3);
   ENABLE3PHASEINVERTER();

   for(;;) {
```

Wait for Switch5 to be On

```
for(;;) {
  __RESET_WATCHDOG(); /* feeds the dog */
  LED1_PIN = HALL_1;
  LED2_PIN = HALL_2;
  LED3_PIN = HALL_3;
  while(SW5_PIN)
  {

  }
  TURNOFFTRANSISTORS();
```

Once On, Display and Check Hall Effect Sensors

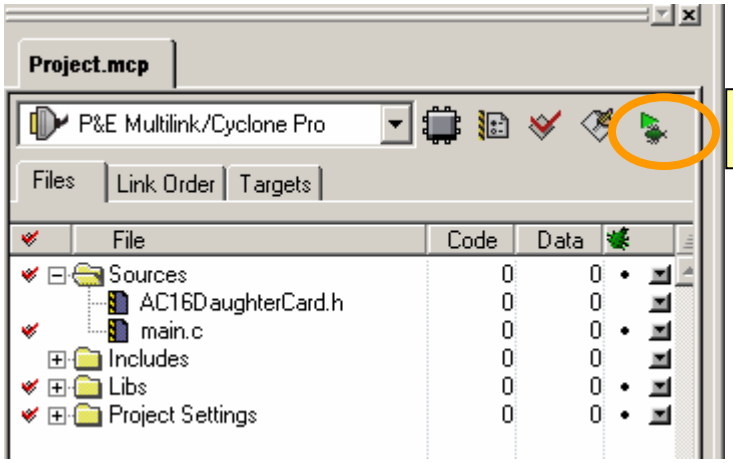
```
while(SW5_PIN)
{
    __RESET_WATCHDOG(); /* feeds the dog */
    LED1_PIN = HALL_1;
    LED2_PIN = HALL_2;
    LED3_PIN = HALL_3;
    hallsensors = 0;
    if(HALL_1) hallsensors |= 0x04;
    if(HALL_2) hallsensors |= 0x02;
    if(HALL_3) hallsensors |= 0x01;
    if(pasthallsensors != hallsensors)
    {
     /* Do something */
    }


}
```

Once On, Display and Check Hall Effect Sensors

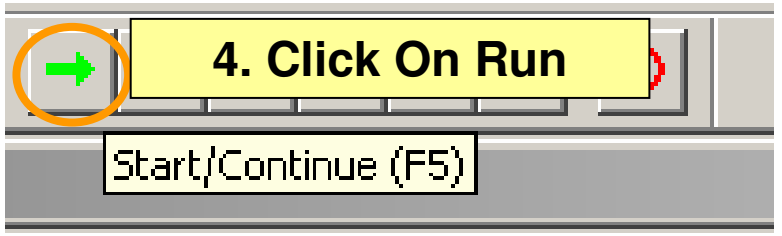
```
if(pasthallhensors != hallsensors)
{
    pasthallsensors = hallsensors;
    value++;
    if(value>=7)  value = 1;
    commutation = table_rotate[value];
    TURNOFFTRANSISTORS();
    if(commutation & Q1_MASK) Q1 = 1;
    if(commutation & Q4_MASK) Q4 = 1;
    if(commutation & Q2_MASK) Q2 = 1;
    if(commutation & Q5_MASK) Q5 = 1;
    if(commutation & Q3_MASK) Q3 = 1;
    if(commutation & Q6_MASK) Q6 = 1;

}
```

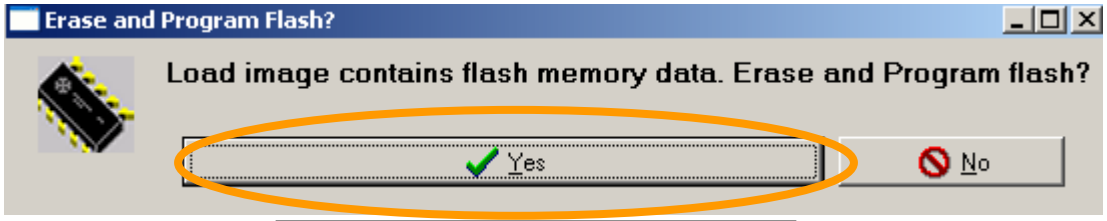
IT IS IMPORTANT TO TURN OFF TRANSISTORS!

Download and Run Code

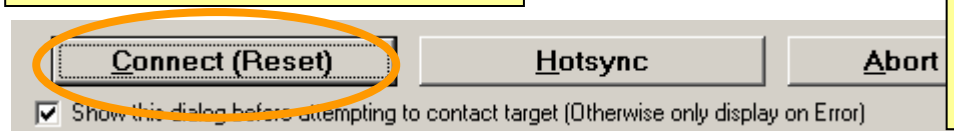


1. Click On Run





2. Click On Connect on the Debugger



3. Click On Yes To Reprogram the MCU

freescale semiconductor

Sensorless Sensing

- Sensors are expensive and take up space
- Several techniques can be used to determine the motor position/speed without an external device
- These techniques are based on the electrical characteristics of motors, mainly on their inductance characteristics:

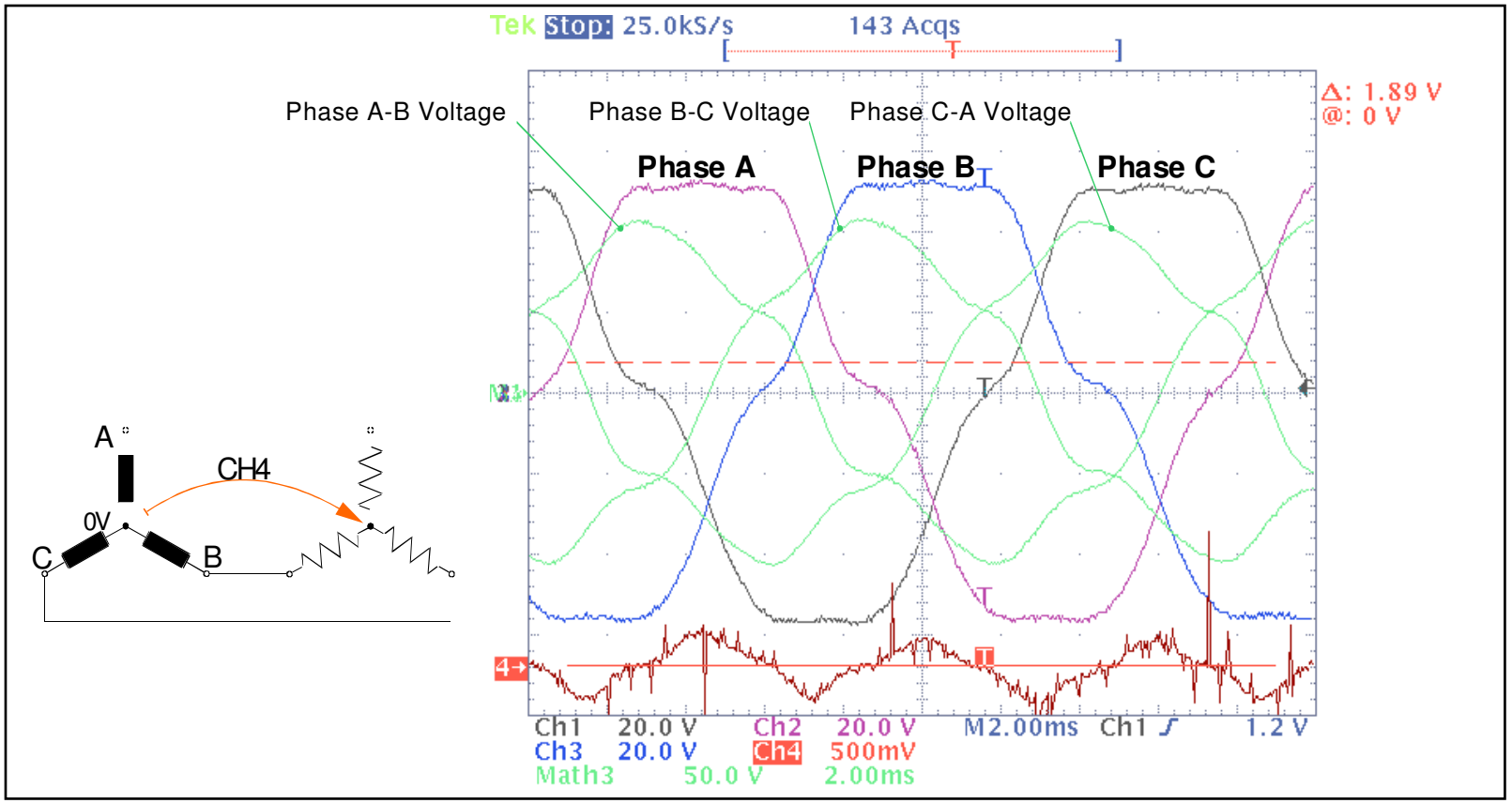
- Based on BEMF
  - Speed range from 5-10% up to 100% of nominal speed
    - The BEMF must be high enough

- Based on Motor Inductance Saliency
  - Speed range from standstill to about 20% of nominal speed

BEMF

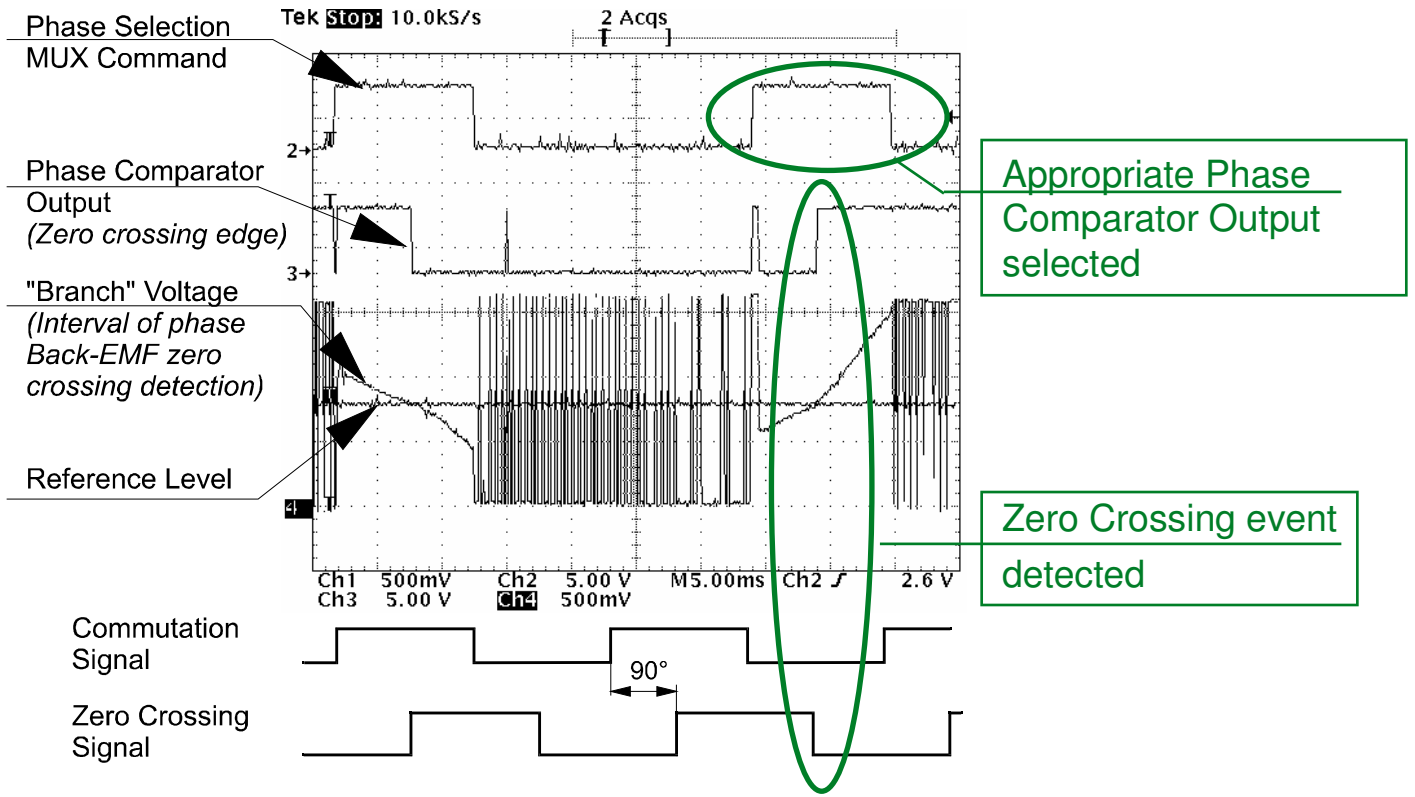
- BEMF is just an acronym for Back Electromagnetic Force
- Back electromagnetic force is a fancy term for the generator characteristics of a motor
- As has been shown not all phases of the motor are on at the same time
- BEMF voltage can be measured on the inactive phases of the motor
- The characteristics of the voltage curve generated by BEMF can tell the position/speed of the motor
- The method that will be exposed is the zero crossing method. When BEMF voltage equals zero, the motor is in a specific position
- By measuring the zero crosses against time, the speed of the motor can be determined

BLDC Motor Back-EMF Shape



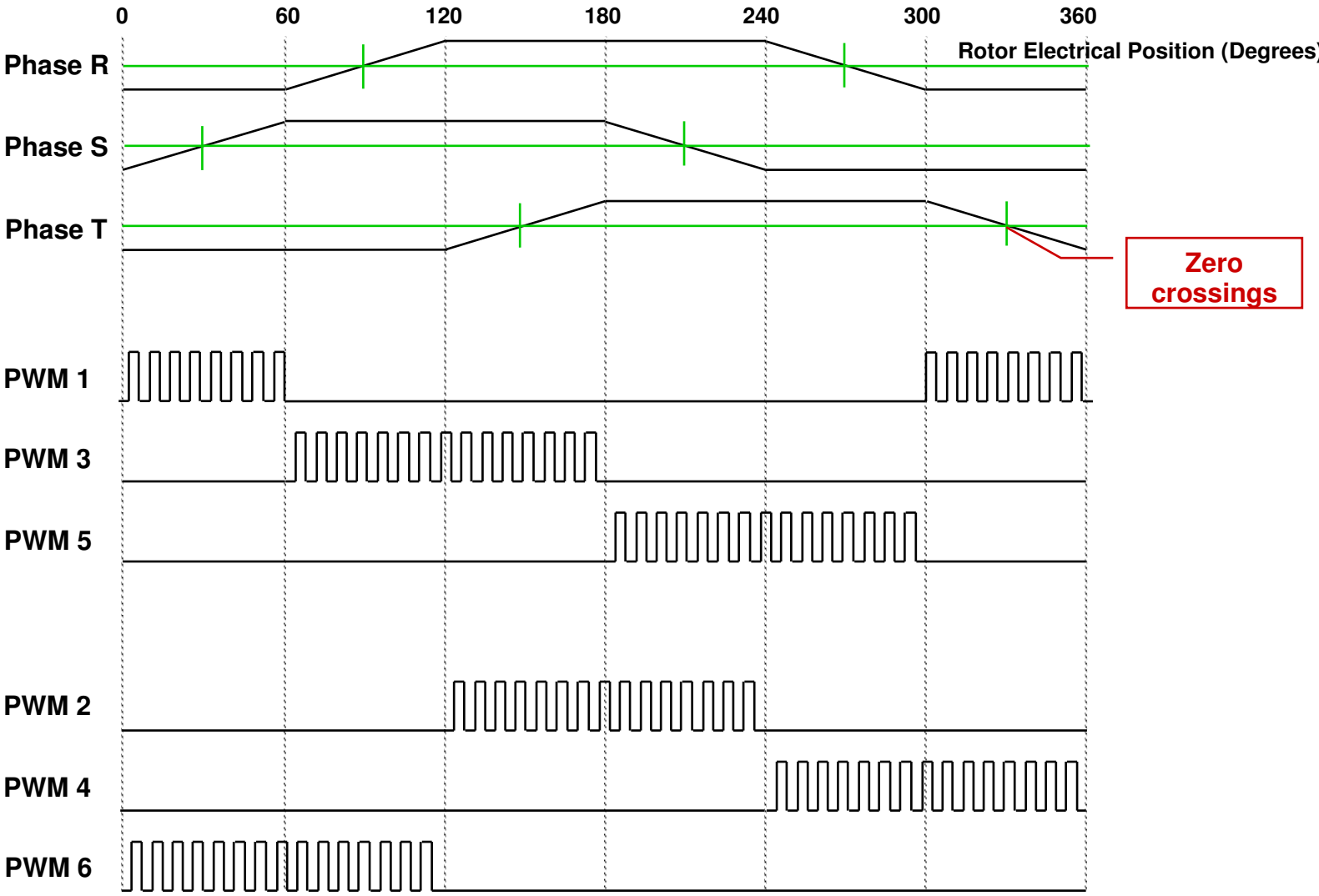
freescale semiconductor

Sensorless BLDC Motor Control with BEMF Zero-Crossing Detection



freescale semiconductor

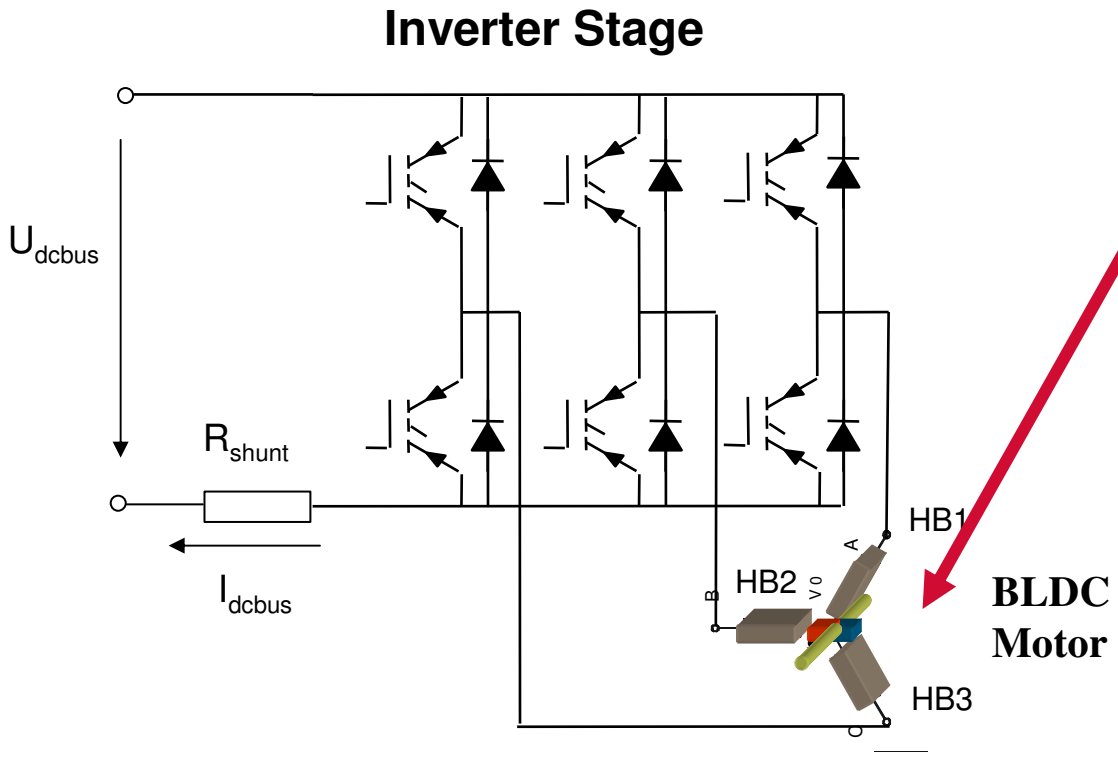
Sensorless Commutation and BEMF



freescale semiconductor

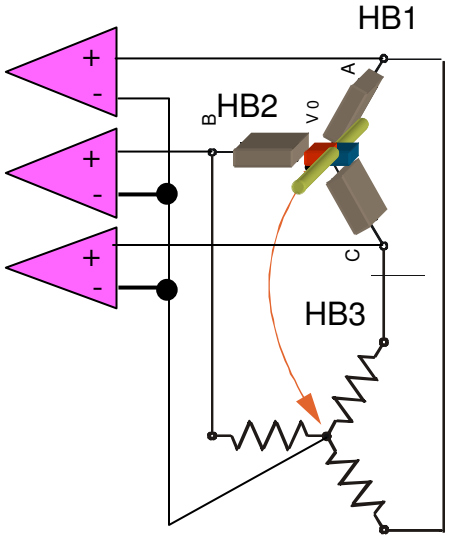
BLDC Central Point is Not Accessible

- 3-phase inverter and DC bus current measurement

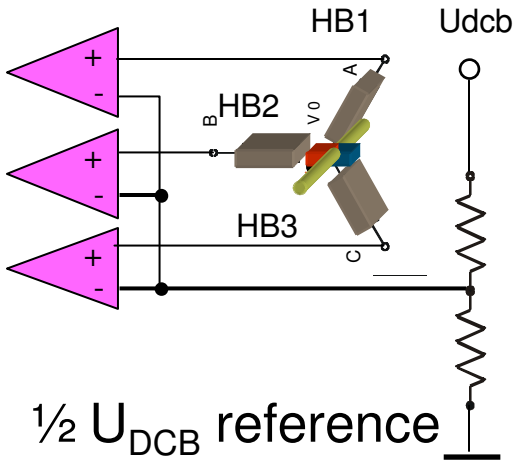


freescale™ semiconductor

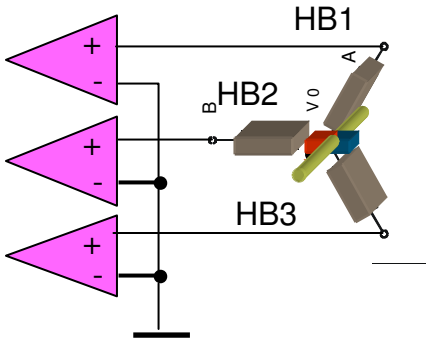
Zero Crossing Sensing Reference



- Virtual CP reference

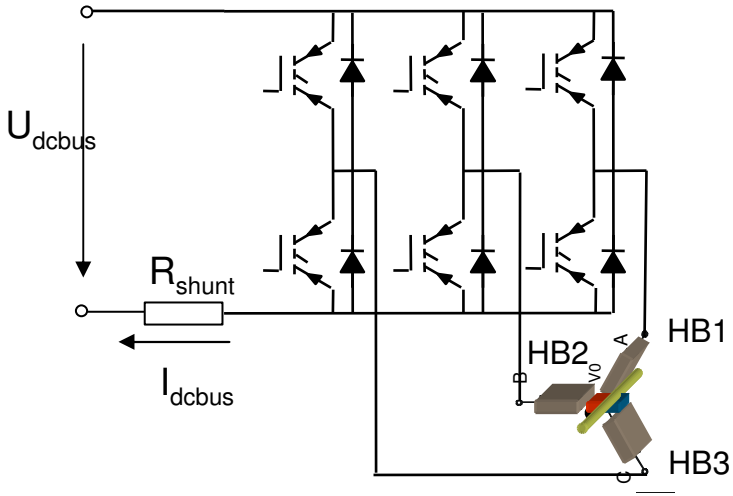


- ½ U_{DCB} reference



- GND reference

- BLDC Motor central point is not accessible



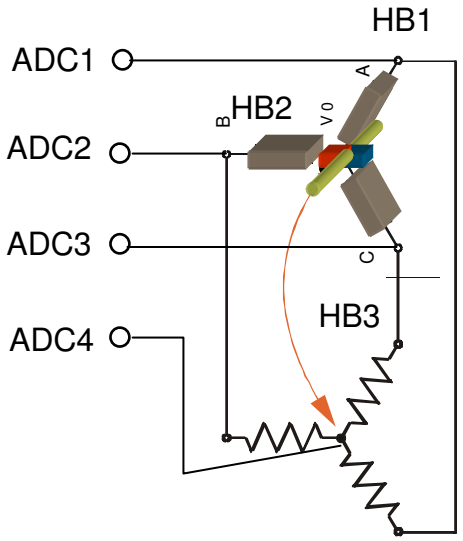
BLDC Motor

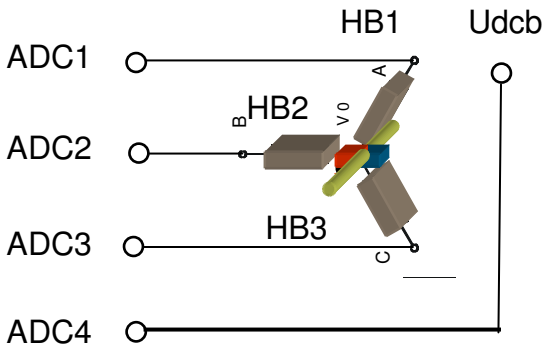
freescale semiconductor

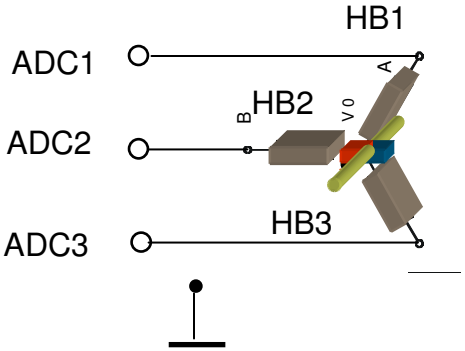
Zero Crossing Sensing using ADC

- The principle is the same as the HW topology, but more flexible

- Virtual CP reference
- **½ U_{DCB} reference**
- GND reference

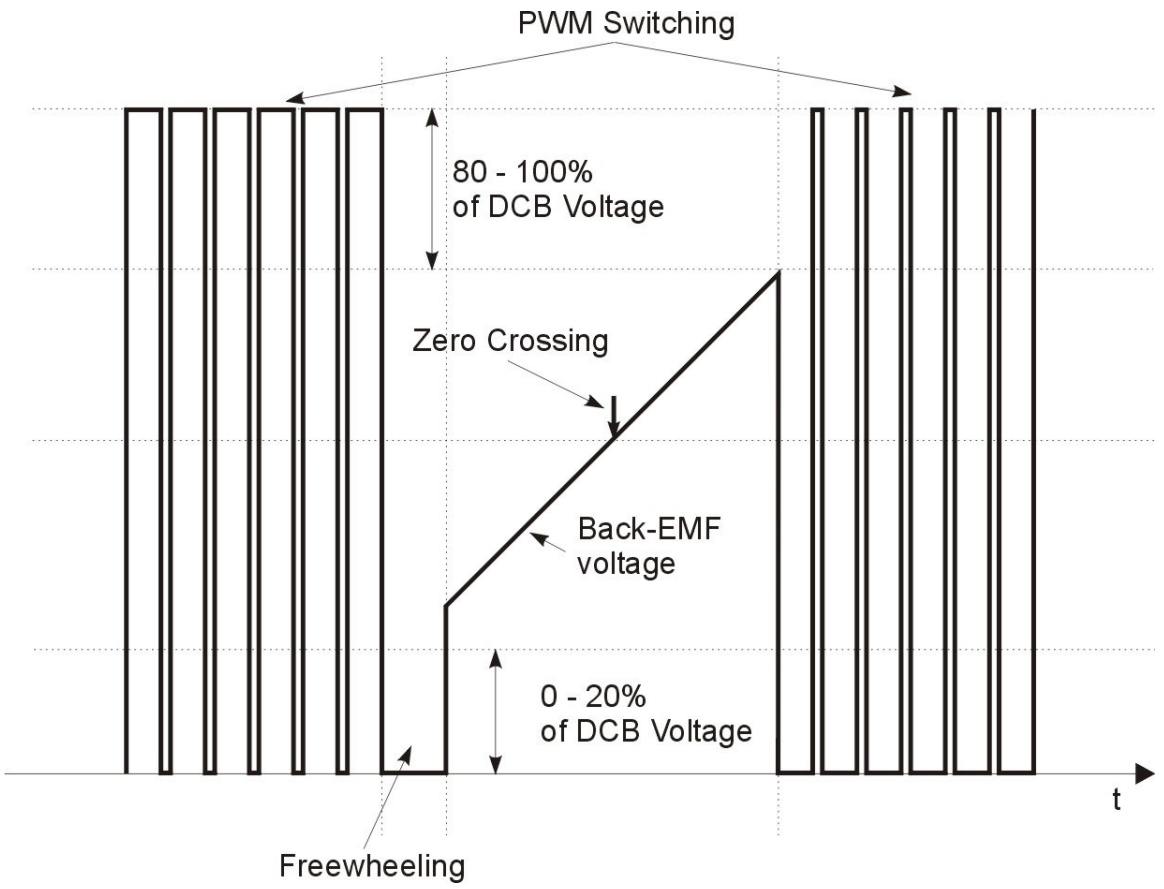




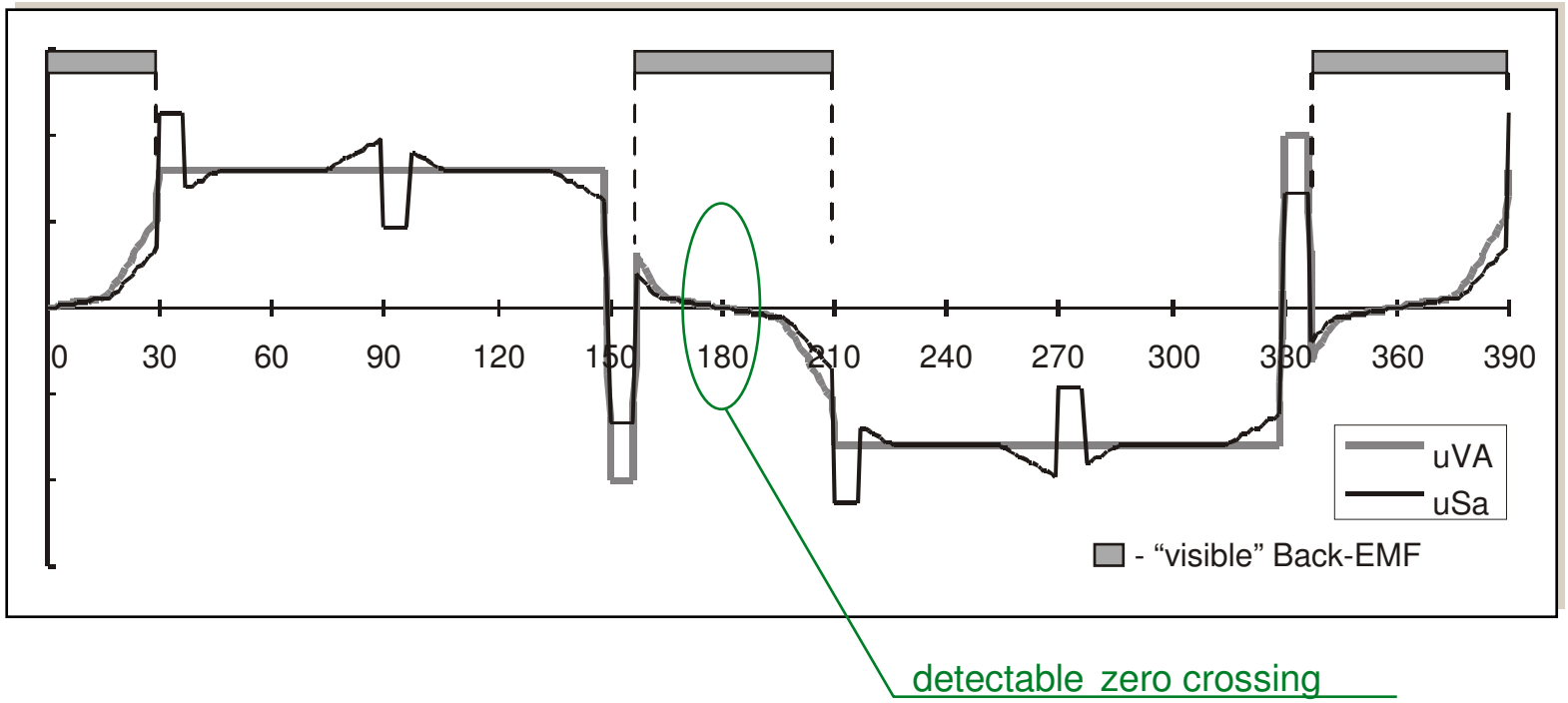


freescale™ semiconductor

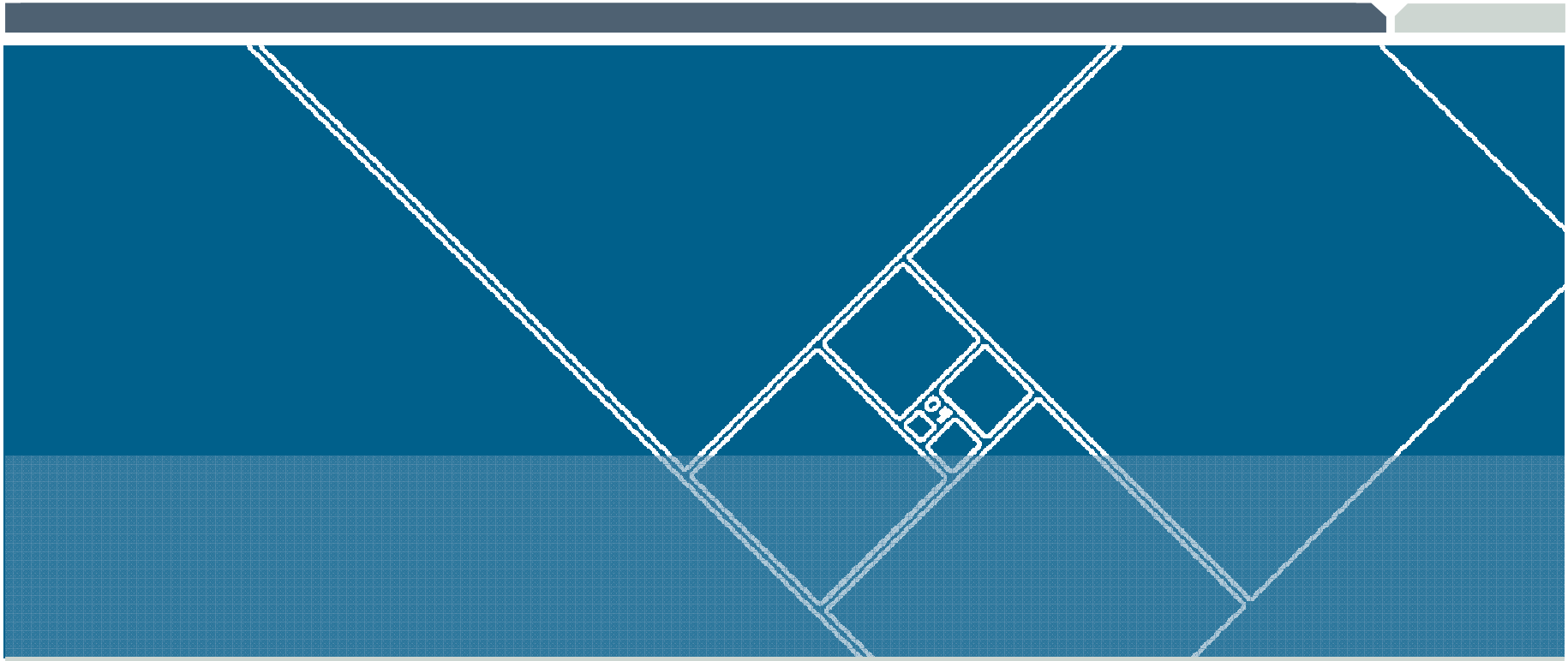
- ADC Measurement – Back-EMF evaluation

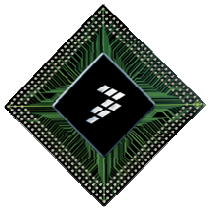


Back-EMF Detection Window



freescale™ semiconductor



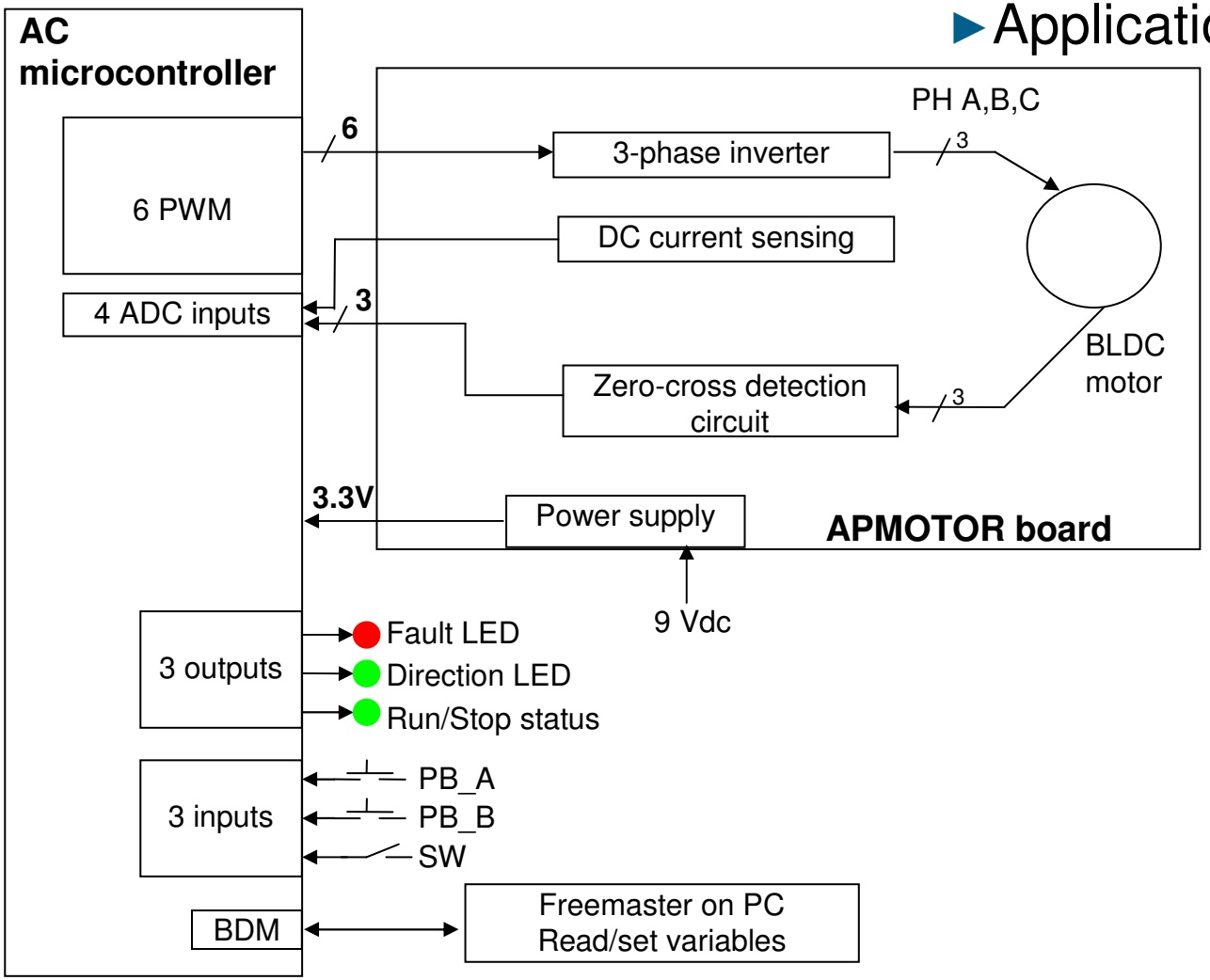


Lab3

freescale™ semiconductor

Sensorless BLDC Motor Control using MC9S08AC

Application Diagram



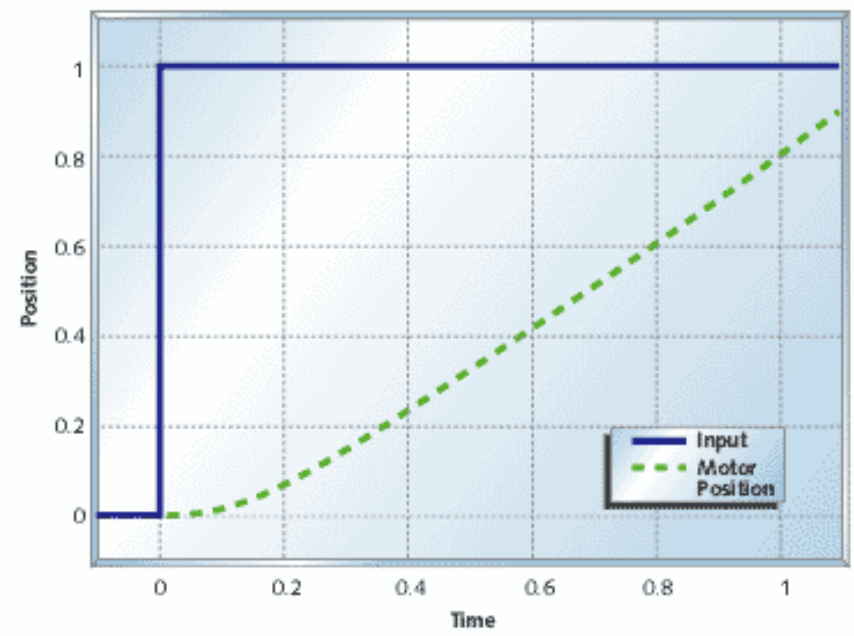
freescale™ semiconductor

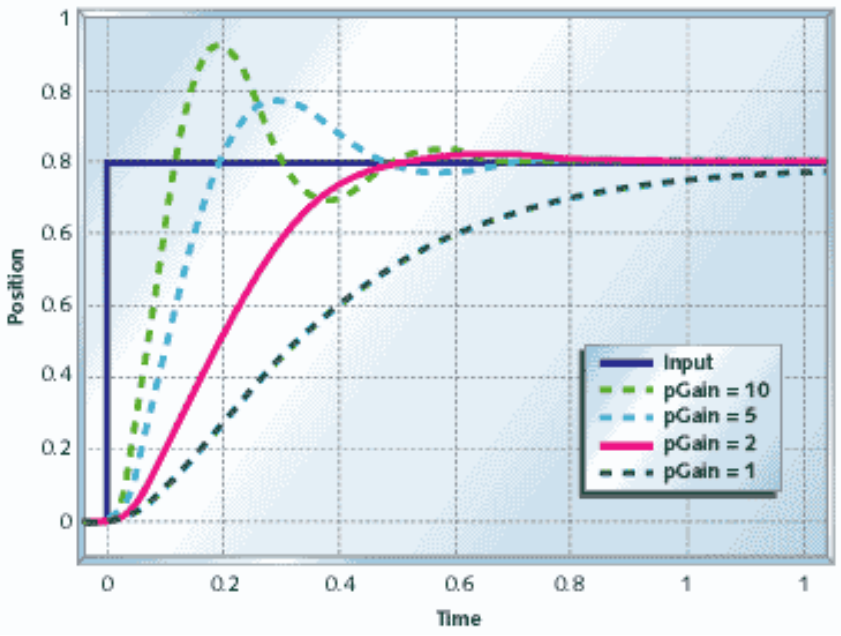
Sensorless BLDC Motor Control using MC9S08AC

- MC9S08AC Peripheral Utilization
  - Timer 1
    - 6 channels: PWM modulation for BLDC motor (complementary bipolar)
  - Timer 2
    - Time base for commutation period measurement
    - Channel 0: commutation
    - Channel 1: timing of application
  - A/D Converter
    - DC bus current, phase voltages (zero-cross detection)

PI Control

- Proportional Control
  - Error multiplied by constant
  - Deals with present behavior

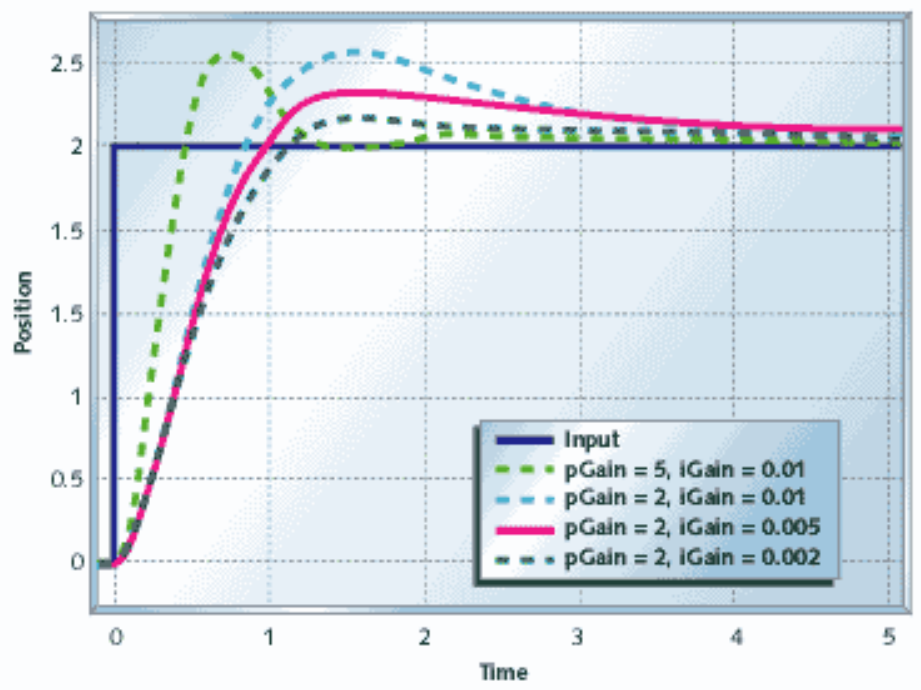




freescale semiconductor

PI Control

- Integral Control
  - Ads long-term precision
  - Takes longer to settle, but provides better precision
  - Deals with past behavior



freescale™ semiconductor

PI controller on AC MCU

```
    if (timer_control == 0) {      /* When timer_control varia

    error = (speed_sp - speed) / error_div;    /* This is the P

    if (error > max_error) error = max_error;      /* If error
    if (error < -max_error) error = -max_error;    /* If error
    pwm_out = pwm_out + error;

    if (pwm_out > pwm_modulo) pwm_out = pwm_modulo;  /* If pwm
    if (pwm_out <= pwm_min) pwm_out = pwm_min;   /* If pwm_out

    pwm_value = pwm_out;
    timer_control = timer_control_val;
}
```

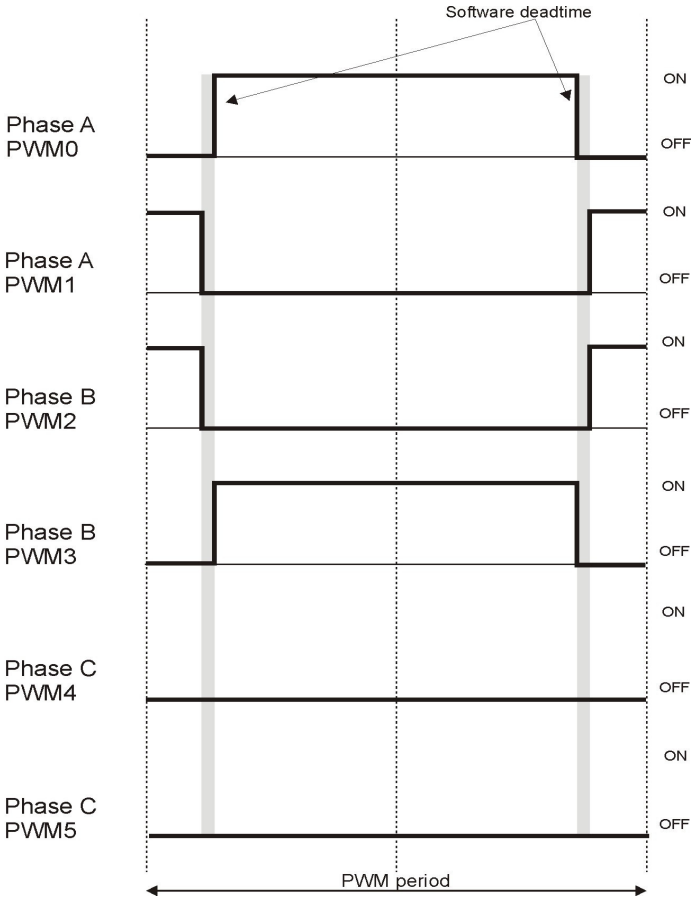
These variables are used To tune the system

This is the control Algorithm inplemented In AC16 MCU for Motor control

freescale semiconductor

PWM and manual dead time insertion

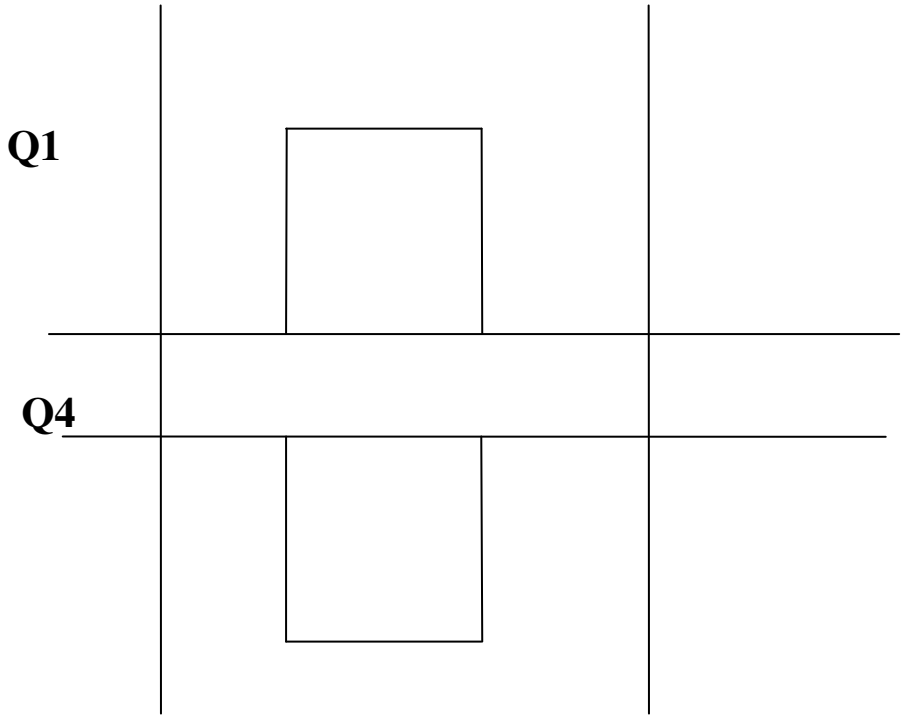
- PWM Generation



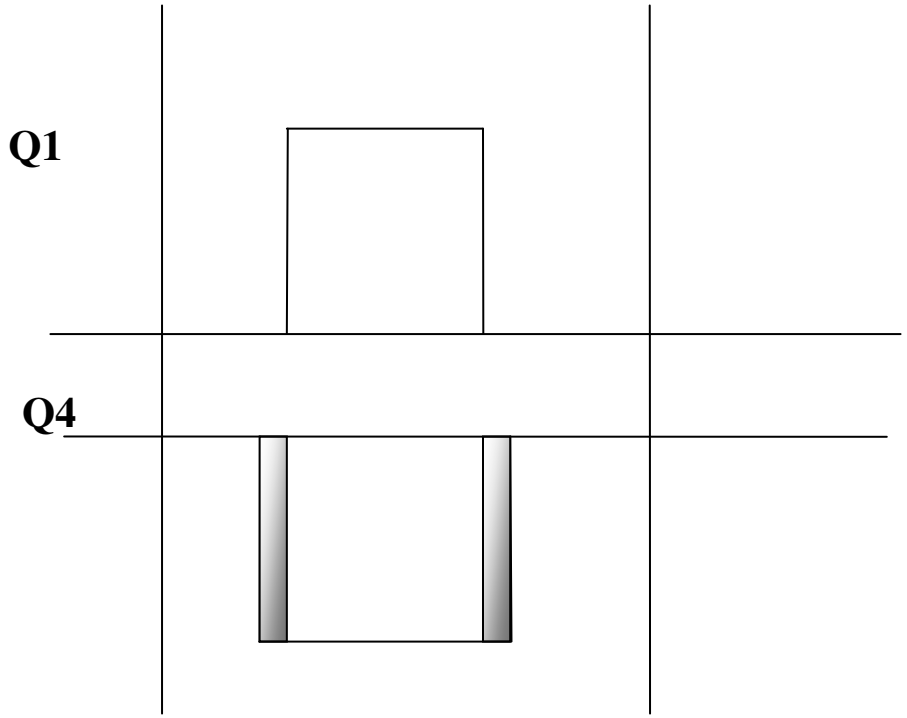
- TIMER set to center aligned mode (TPM1SC:CPWMS=1)
- Example:
  - PWM0: switching (duty cycle (50 - 100%) + dead time), negative polarity (TMP1CxSC:ELSnB =x,TMP1CxSC:ELSnA =1)
  - PWM1: switching (duty cycle (50 - 100%) - dead time), positive polarity (TMP1CxSC:ELSnB =1,TMP1CxSC:ELSnA =0)
  - PWM2: switching (duty cycle (50 - 100%) - dead time), positive polarity (TMP1CxSC:ELSnB =1,TMP1CxSC:ELSnA =0)
  - PWM3: switching (duty cycle (50 - 100%) + dead time), negative polarity (TMP1CxSC:ELSnB =x,TMP1CxSC:ELSnA =1)
  - PWM4: OFF
  - PWM5: OFF

freescale™ semiconductor

Dead Time



a) Center aligned PWM, Q1 and Q4 change in the same instant, it can short circuit between Vb and GND



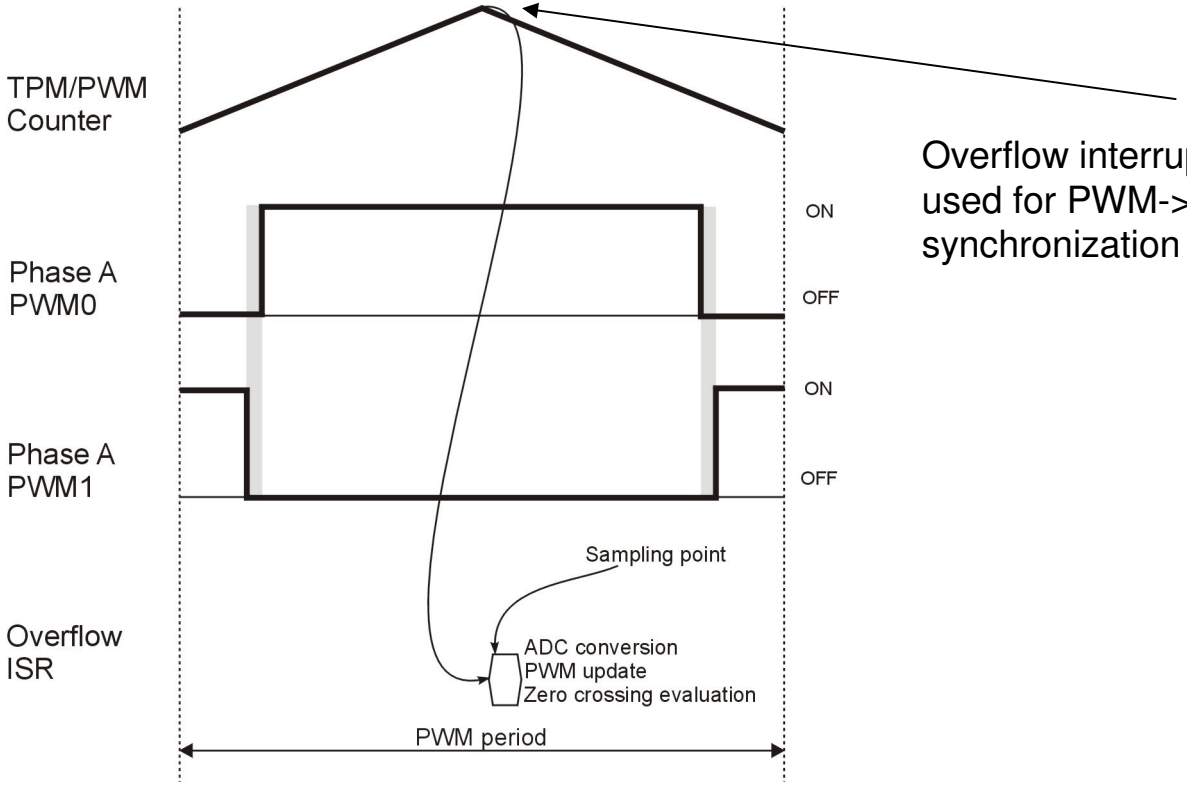
b) Center aligned PWM, Q1 and Q4 triggered with different PWM duty cycle avoiding that both transistors turn on at the same time.

freescale™ semiconductor

Application Details

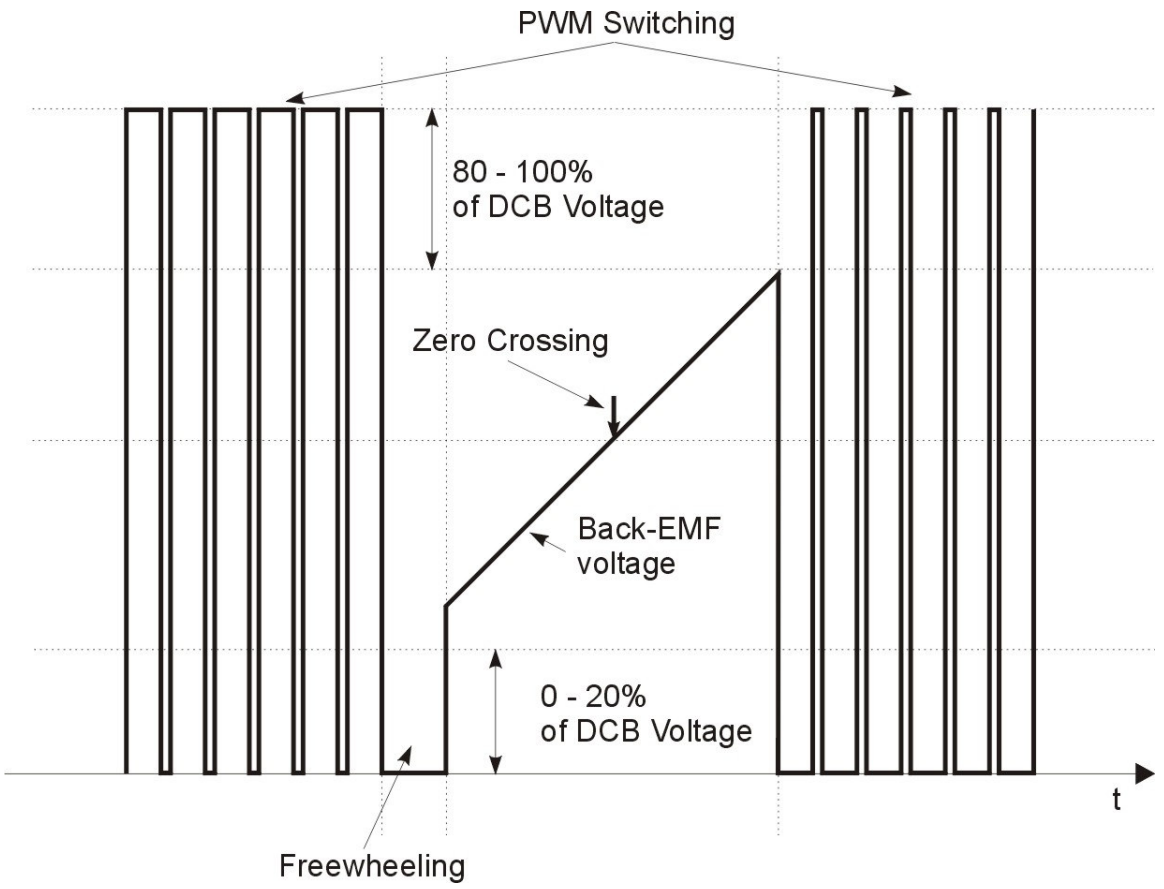
- ADC Measurement
  - DC bus current, Back-EMF voltage
  - Single result register only
  - 3.5 us conversion time
  - ADC measurement has to be synchronized with PWM

- ADC Measurement – PWM -> ADC Synchronization

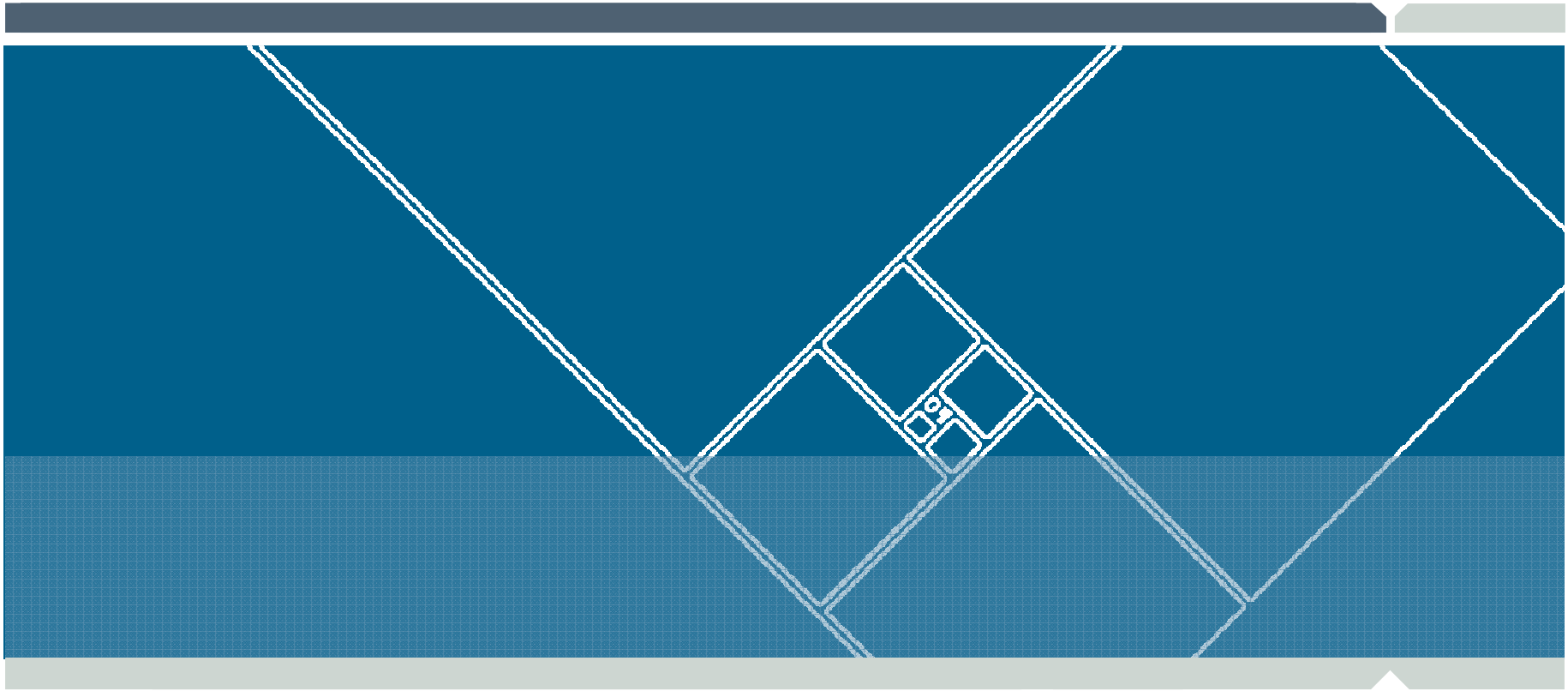


freescale semiconductor

- ADC Measurement – Back-EMF evaluation



freescale™ semiconductor



FlexTimer in the MCF51AC

This part pending because dev board was delivered on April 29

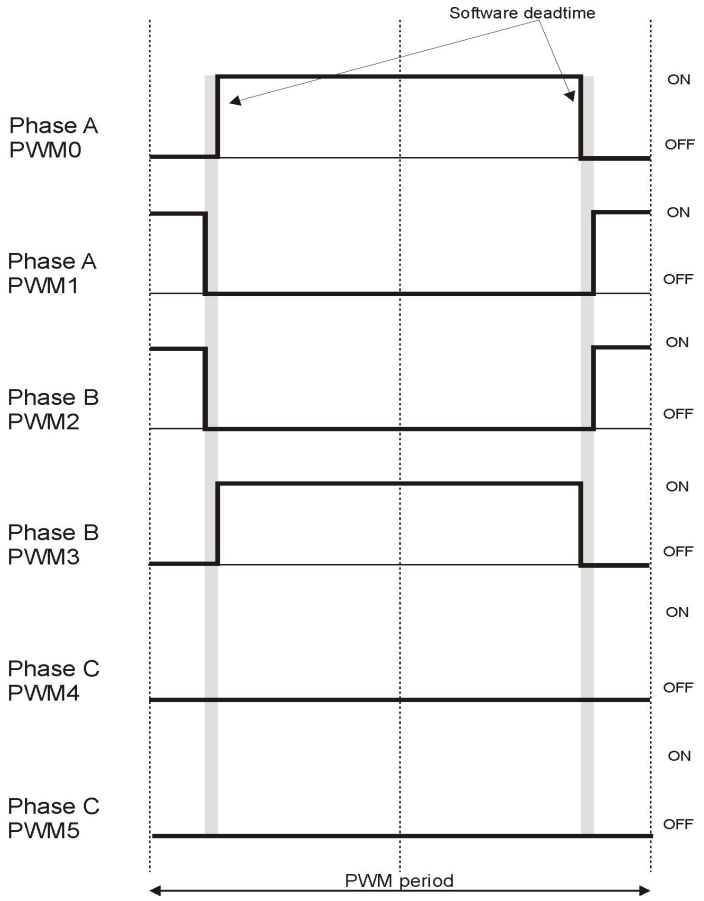
freescale™ semiconductor

FlexTimer advantages

- Supports up to 8 channels which can be synchronized in pairs for complementary signal generation.
- Dead time insertion supported by software.
- FTM can trigger ADC conversions automatically.
- Fault input supported by hardware (automatically turns of PWM pin outputs).
- Synchronized reloading of PWM duty cycle from several sources (ADC, analog comparator, software).
- Polarity for PWM output can be configured.
- Edge and center alligned PWM generation.

Dead time insertion

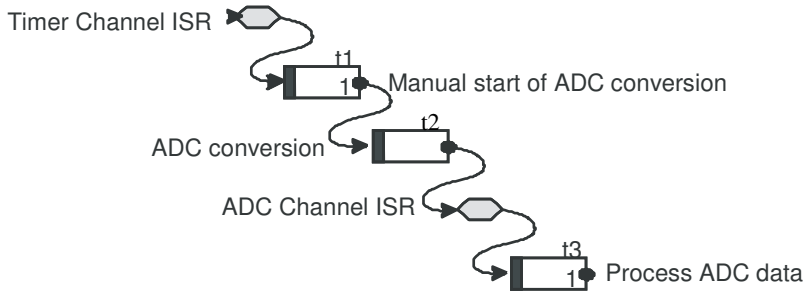
- Used to avoid to power devices to be turned on at the same time.
  - No CPU load generated to make dead time insertion.
  - To configure simply enable dead time insertion bit and configure the number of timer counts of dead time, the rest is done by timer module.



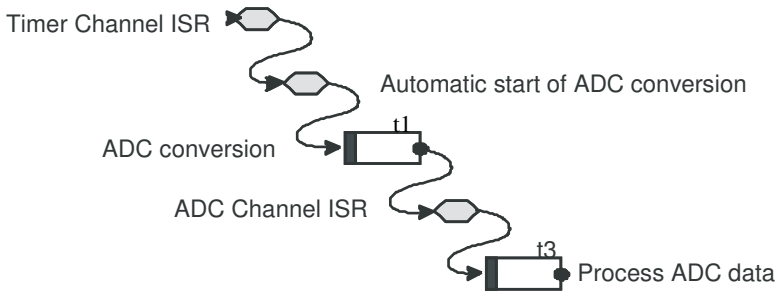
freescale™ semiconductor

ADC Synchronization

- Reduces CPU load by saving time needed to start conversions (to detect zero-crossings or instantaneous current.
- When doing back-EMF sensing measurements need to be made in certain timing windows. If measurements are always taken at the same times, control algorithm is more precise.



- Without hardware trigger
- T = t1 + t2 + t3



With hardware trigger
T = t1 + t3

freescale™ semiconductor

Related Session Resources

Sessions (Please limit to 3)

Session ID	Title
AZ131	Motor Control Part 1 – Fundamentals and Freescale Solutions
AZ121	Motor Control Part 2 – Solutions for Large Appliances and HVAC
AZ120	Motor Control Part 3 – Solutions for Small Appliances and Health Care Applications

Demos (Please limit to 3)

Pedestal ID	Demo Title
312	Large Appliance Demo
214	3-Phase PMSM Vector Control demo with Encoder
706	Air Hockey Demonstration featuring the Flexis AC Products

Meet the FSL Experts (Please limit to 3)

Title	Time	Location

freescale™ semiconductor


freescale™
semiconductor