// Arduino sensored BLDC (brushless DC) motor control code

byte motor\_speed, bldc\_step, pwm\_pin = 2;

void setup() {

  DDRD  |= 0xFC;                     // 2, 3, 4, 5, 6 ve 7 pinlerini çıkış olarak yapılandırın

  PCICR  = 1;                        // 8 ila 13 numaralı pinler için pin değiştirme kesintisini etkinleştirin

  PCMSK0 = 7;                        // 8, 9 ve 10 pinleri için pin değiştirme kesintisini etkinleştirin

  // Timer2 yapılandırması

  TCCR2A = 0;

  TCCR2B = 2;                        // Timer2 prescaler = 1/8 (Timer2 clock = 2MHz)

  TCNT2  = 0;                        // Timer2 preload value = 0

  // ADC module configuration

  ADMUX  = 0x60;                     // // ADC modülünü yapılandırın ve 0 kanalını seçin

  ADCSRA = 0x84;                     // // 16 bölme faktörlü ADC modülünü etkinleştirin (ADC saati = 1MHz)

  // BLDC motor first move

  bldc\_step = PINB & 7;              // Hall etkisi sensörlerinin durumunu okuyun (PINB: PORTB'den okuyun) arduino pins 8..13)

  bldc\_move();                       // BLDC motorunu hareket ettirin (ilk hareket)

}

ISR (PCINT0\_vect){

  bldc\_step = PINB & 7;              // Hall etkisi sensörlerinin durumunu okuyun ve kaydedin (PINB: arduino pinleri 8..13 olan PORTB'den okuyun)

  bldc\_move();                       // BLDC motorunu hareket ettirin

}

void bldc\_move(){                    // Hall etkisi sensörlerinin durumuna göre BLDC motor hareket fonksiyonu

  switch(bldc\_step){

    case 1:

      PORTD = 0x48;

      pwm\_pin = 2;

      break;

    case 2:

      PORTD = 0x24;

      pwm\_pin = 4;

      break;

    case 3:

      PORTD = 0x60;

      pwm\_pin = 4;

      break;

    case 4:

      PORTD = 0x90;

      pwm\_pin = 6;

      break;

    case 5:

      PORTD = 0x18;

      pwm\_pin = 2;

      break;

    case 6:

      PORTD = 0x84;

      pwm\_pin = 6;

      break;

    default:

      PORTD = 0;

      break;

  }

}

void loop() {

  ADCSRA |= 1 << ADSC;              // Dönüşümü başlat

  while(ADCSRA & 0x40);              / /Dönüşümün tamamlanmasını bekleyin

  motor\_speed = ADCH;                // ADC verilerini oku (8 bit)

  // Software PWM

  if(motor\_speed < 250){

    while(TCNT2 < motor\_speed) ;

    digitalWrite(pwm\_pin, LOW);

  }

  if(motor\_speed > 0){

    while(TCNT2 >= motor\_speed) ;

    digitalWrite(pwm\_pin, HIGH);

  }

}