```
void setup() {

  pinMode(Hu, INPUT_PULLUP);  // set input mode with pull-up MOS for hall sensor
  pinMode(Hv, INPUT_PULLUP);
  pinMode(Hw, INPUT_PULLUP);

  attachInterrupt(Hu, HuIrq, CHANGE);  // attach interrupt
  attachInterrupt(Hv, HvIrq, CHANGE);
  attachInterrupt(Hw, HwIrq, CHANGE);

  setPWM();  // set 3-phase PWM

}

//モータ制御, 周期10ms
void loop() {
  static int mode, count = 0, mod;
  float accel, duty, Vdc;
  static float prev_duty;
  long time_s;

  switch (mode) {  //始動処理, アクセルが戻ったかを確認
    case STOP:
      if (R_ADC0->ADDR[9] < ACCEL_OFFSET) {
        mode = RUN;  //走行モードへ
        dir = 0;
        pulseOutput();
      }
      break;
    case RUN:                   //走行モード
      accel = R_ADC0->ADDR[9];  //アクセルデータを取得

      accel = accel - ACCEL_OFFSET;                       //アクセル値からDutyへの変換
      if (accel <= DED_TIME) accel = DED_TIME + 1;     //負値のリミット処理
      if (accel > ADC_RESOLUTION) accel = ADC_RESOLUTION; //分解の以上のリミット処理
      accel = (5.0 / 3.3) * accel * PWM_CYCLE / (ADC_RESOLUTION - ACCEL_OFFSET);  //PWMカウントへ変換
      if (accel >= (PWM_CYCLE - DED_TIME)) accel = PWM_CYCLE - DED_TIME; //PWM周期カウント範囲以上のリミット処理
      duty = accel / ADC_RESOLUTION * 100 * 5.0 / 3.3;
      if(duty >= 100) duty = 100;
      if(duty <= 0) duty = 0;

      R_GPT2->GTCCR[2] = R_GPT3->GTCCR[2] = R_GPT7->GTCCR[2] = DED_TIME + (PWM_CYCLE - DED_TIME) * (100 - duty) / 100;  //

      if ((R_ADC0->ADDR[9] < (ACCEL_OFFSET / 2))  //アクセル値が100/1023の場合は起動モードへ
        mode = STOP;                                                    //停止モードへ
        count = 0;                                                      //送信の周期を初期化
        duty = 0;                                                       //dutyを0%
      }
  }

  delay(10);
  pwmT.pulse_perc(R_ADC0->ADDR[0]*100.0/ADC_RESOLUTION);
}
```