

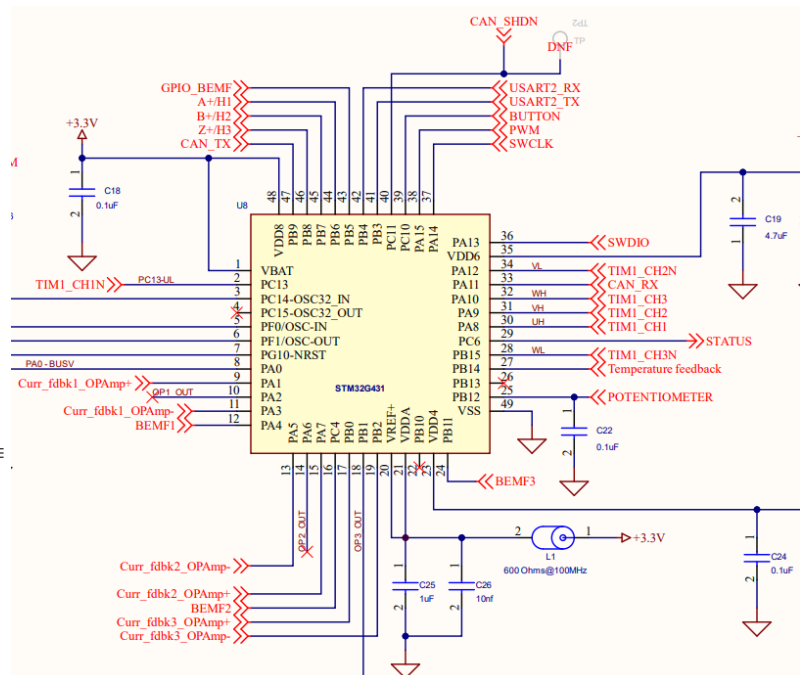
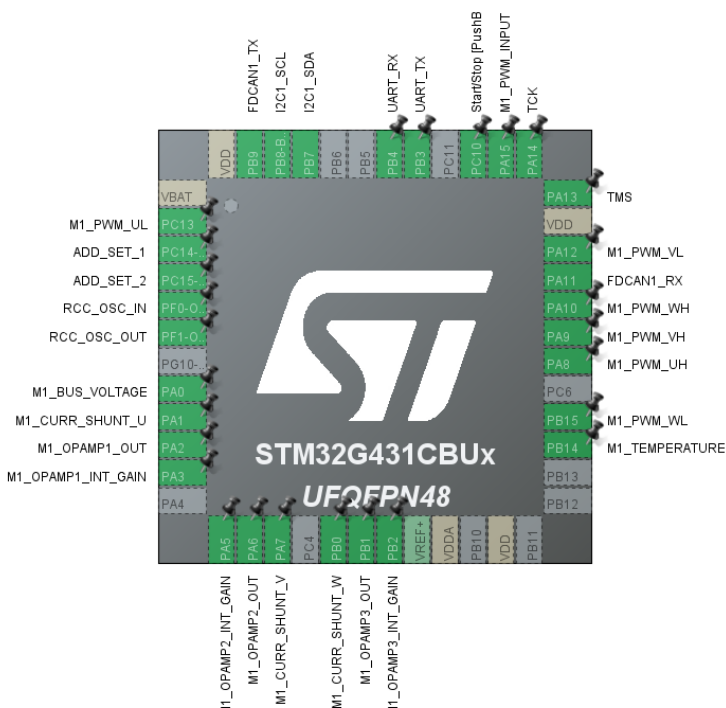
# AM32 偵測轉子方法

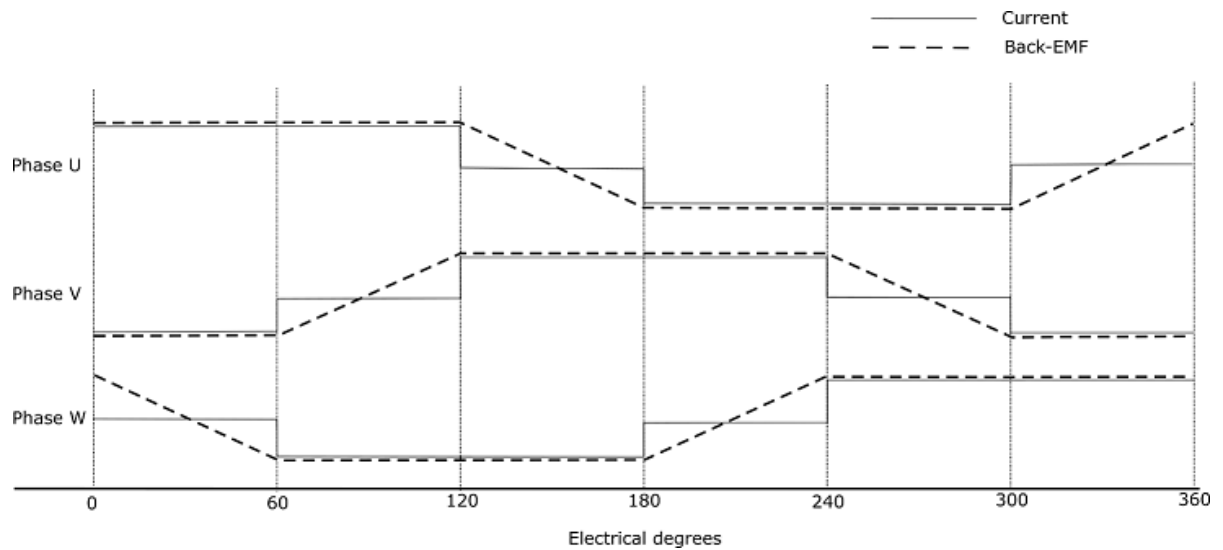
採用**方波無感驅動**（梯形波換相），其轉子位置的推算主要依賴於**反電動勢**（Back-EMF, **BEMF**）訊號的監測，具體而言，AM32 在每一次換相時會讓一相線圈處於斷路狀態，透過 ADC 模擬輸入讀取該浮空相的電壓，檢測 **BEMF 過零點**來推斷轉子位置和定時換相。AM32 提供**正弦波起動模式**來改善啟動時的順暢度與扭力，實際上是在低速時以近似 FOC 的方式驅動，待轉子建立足夠 BEMF 後再切換回方波模式

## STM32G431 (B-G431B-ESC1) 平台的常見腳位功能配置

- **三相 PWM 輸出（Gate 驅動）**：使用 STM32G431 的 TIM1 高速 PWM 通道來驅動三相全橋的 6 個開關。
  - Phase A：高邊 PWM 輸出腳為 PA8 (TIM1\_CH1)，低邊 PWM 輸出腳為 PC13 (TIM1\_CH1N)。
  - Phase B：高邊 PWM 為 PA9 (TIM1\_CH2)，低邊 PWM 為 PA12 (TIM1\_CH2N)。
  - Phase C：高邊 PWM 為 PA10 (TIM1\_CH3)，低邊 PWM 為 PB15 (TIM1\_CH3N)。這些 PWM 腳位經過板上 L6387 驅動器放大後控制功率 MOSFET 開通與關斷，實現三相橋臂的換相驅動。
- **BEMF 反電動勢感測（無感位置估計）**：為實現無感轉子位置檢測，三相各有一路模擬輸入接到相電樞節點，用於測量反電動勢電壓：
  - BEMF1：PA4 引腳，接相位 A 的反電動勢感測。
  - BEMF2：PC4 引腳，接相位 B 的反電動勢感測。
  - BEMF3：PB11 引腳，接相位 C 的反電動勢感測。

在 AM32 中，三路 BEMF 電壓透過 ADC 交替讀取，用於檢測浮空相電壓的**過零點**。當某一相 BEMF 越過虛擬中性點電壓時，即判定轉子磁極通過定子線圈中點，據此觸發下一次換相定時。





#### (A) STM32 FOC + STO PLL → 正弦波 FOC 全閉迴圈控制

(官方 MCSDK / 你之前用 G431 工具產生的 FOC 工程)

控制流程是這樣：

```
java
複製 編輯

讀取 IA、IB (兩相電流)
↓
Clarke 變換 → 變成 αβ 座標
↓
State Observer (STO) + PLL → 推算磁鏈角度 (轉子位置θ)
↓
Park 變換 → 變成 dq 座標 (定子同步旋轉座標)
↓
PI 控制器調整 Iq, Id → 決定馬達力矩/磁通
↓
反向 Park → 變回 αβ 電壓指令
↓
SVPWM or Space Vector → 控制 PWM 輸出三相波形
↓
馬達驅動
```

- ✓ 關鍵：每一次 PWM 更新時都重新計算「轉子角度 $\theta$ 」→ 正弦 FOC 驅動
- ✓ 轉子角 $\theta$ 來源 → 是 STO + PLL 算出來的 → 只用 IA/IB 電流 + Vbus，不用 BEMF ADC
- ✓ 特色：完全 FOC，馬達電流波形是平滑正弦波，效率高，低速表現好

#### (B) AM32 → 六步換相 + BEMF Zero Crossing 驅動

AM32 用的是「六步方波驅動」(Trapezoidal Control)，流程是：

```
pgsql
複製 編輯

起動階段 (低速時) → sine startup or open loop commutation
↓
進入正常運轉 → Zero-Crossing Detection (ZCD)

主迴圈：
  讓其中一相浮空
  ↓
  讀取該相 BEMF 電壓 → 等待過零
  ↓
  檢測到過零點 → 計算下一步換相定時 (commutation timing)
  ↓
  切換 PWM 控制六步換相表 → 換到下一組 PWM 驅動狀態
  ↓
  重複
```

- ✓ 關鍵：並不是 FOC → 沒有 Park / Clarke 變換，也沒有 dq 軸控制
- ✓ 轉子位置只在「換相」時才更新一次 → ZCD 得到目前在某一區段
- ✓ 控制的是六步表，不是正弦電流

## 參考資料

- AM32 官方 GitHub 倉庫：<https://github.com/am32-firmware/AM32>
- STM32 Motor Control SDK 的六步換相算法說明：  
[https://wiki.stmicroelectronics.cn/stm32mcu/wiki/STM32MotorControl%3A6-step Firmware Algorithm](https://wiki.stmicroelectronics.cn/stm32mcu/wiki/STM32MotorControl%3A6-step+Firmware+Algorithm)