

# ロータリエンコーダについて

今回はモータと一体型のエンコーダを使用します。(DMN29BM)  
オープンコレクタタイプとなっていますのでV<sub>cc</sub>とセンサ出力の間にプルアップ抵抗が必要です。

オープンコレクタ回路とはトランジスタのコレクタを内部でどこにも接続せず、そのまま端子に引き出したものです。

```
• • •  
void DeviceInitFunc(void){  
• • •  
    INTCON2 = 0;  
    IEC0bits.INT0IE = 1; /*Enable INT0 Interrupt Service Routine */  
    IEC1bits.INT2IE = 1; /*Enable INT2 Interrupt Service Routine */  
• • •  
}  
• • •
```

外部割り込み有効

```
• • •  
static volatile int MotorRotat[2] = {0,0};  
  
int GetMotorRotat(const short MotorType){  
    int MotRotat;  
    MotRotat = MotorRotat[MotorType];  
    MotorRotat[MotorType] = 0;  
  
    return MotRotat;  
}  
• • •
```

モータが何回転したのか  
を返す関数です。

```

• • •
void __attribute__((__interrupt__)) _INT2Interrupt(void)
{
    static int EncPulseCount = 0;
    if((ENC_2A == 1 && ENC_2B == 1) || (ENC_2A == 0 && ENC_2B == 0)){
        EncPulseCount++;
    }else{
        EncPulseCount--;
    }
    if(EncPulseCount >= 12){
        EncPulseCount = 0;
        MotorRotat[motor2]++;
    }else if(EncPulseCount <= -12){
        EncPulseCount = 0;
        MotorRotat[motor2]--;
    }
    IFS1bits.INT2IF = 0;
}
• • •

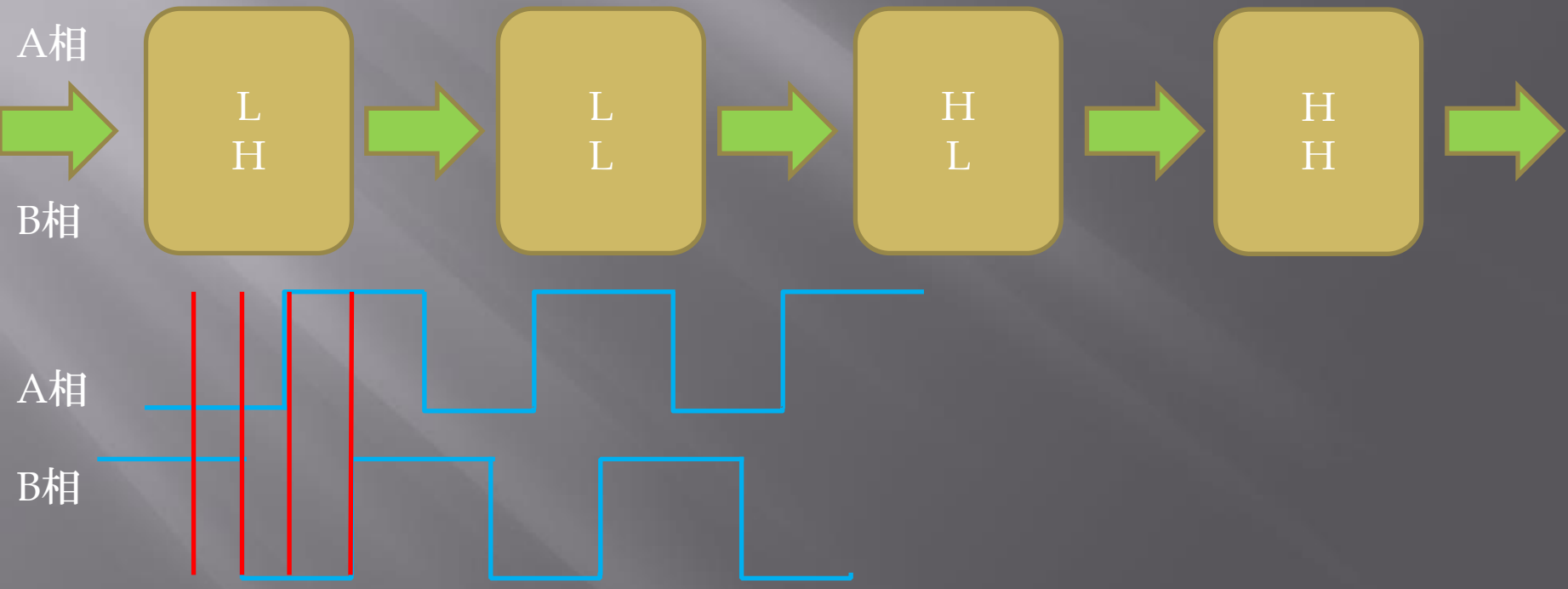
```

外部割り込み関数。  
エンコーダのパルスのカウントし、  
モータ回転数を計測する。  
詳しい解説は後ほど

# 回転数カウント方法の詳細

割り込み関数内では12パルスを1回転としています。これはエンコーダの使用によるものです。  
次にCW,CCWの判定方法ですが以下のような法則性があります。

high,lowの組み合わせは4パターンしか存在しません。

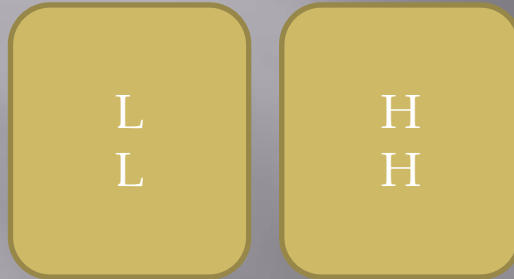


## 回転数カウント方法の詳細(2)

B相が変化した時に外部割り込みがかかります。つまり

## CWの際は

の2パターンしか存在しません



## CCWの際は

の2パターンしか存在しません

この関係性は初期値に依存せず  
常に不変です。



# エンコーダの回路図

