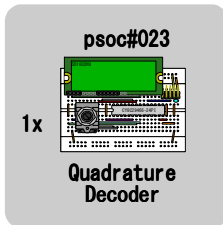




15+

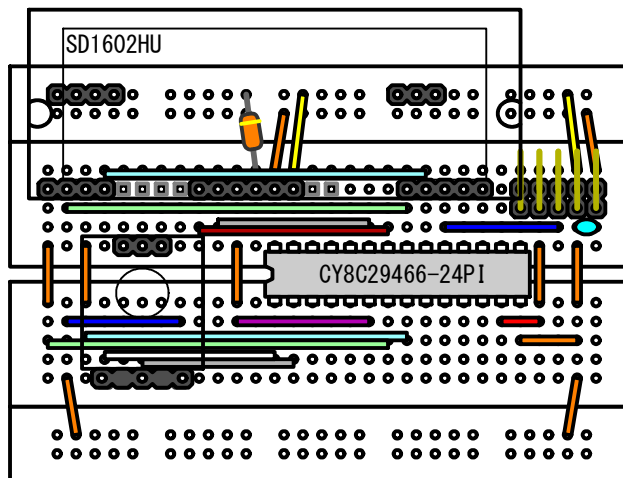
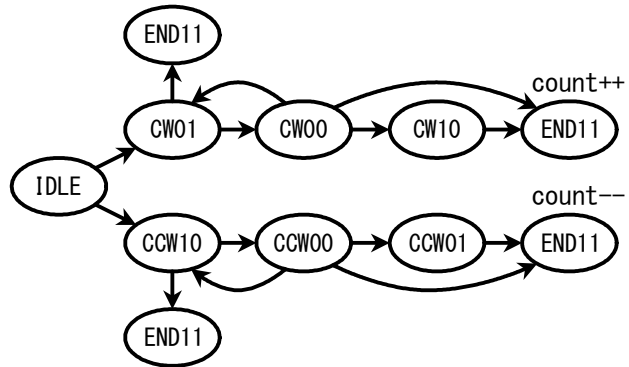
025

Quadrature Decoder
by GPIOInterrupt



PSoCGPIoint.asm

```
PSoC_GPIO_ISR:  
    ljmp _gpio_int_func  
    reti
```



main.c

```
#define RA 0x80 // RA = P0[7]
#define RB 0x20 // RB = P0[5]
enum eSTATE{IDLE,CW01,CW00,CW10,CCW10,CCW00,CCW01,END};
BYTE csw,count=0,stat=IDLE;

#pragma interrupt_handler gpio_int_func
void gpio_int_func(void) {
    while(stat!=END) {
        csw = PRTODR; // Current Switch
        switch(stat) {
            case CW01:
                if ((~csw&RA)&&(~csw&RB)) stat = CW00;
                if ((csw&RA)&&(csw&RB)) stat = END;
                break;
            case CW00:
                if ((csw&RA)&&(~csw&RB)) stat = CW10;
                if ((~csw&RA)&&(csw&RB)) stat = CW01;
                if ((csw&RA)&&(csw&RB)) { stat = END; count++;}
                break;
            case CW10:
                if ((csw&RA)&&(csw&RB)) { stat = END; count++;}
                break;
            case CCW10:
                if ((~csw&RA)&&(~csw&RB)) stat = CCW00;
                if ((csw&RA)&&(csw&RB)) stat = END;
                break;
            case CCW00:
                if ((~csw&RA)&&(csw&RB)) stat = CCW01;
                if ((csw&RA)&&(~csw&RB)) stat = CCW10;
                if ((csw&RA)&&(csw&RB)) { stat = END; count--;}
                break;
            case CCW01:
                if ((csw&RA)&&(csw&RB)) { stat = END; count--;}
                break;
            case IDLE:
                if ((~csw&RA)&&(csw&RB)) stat = CW01;
                if ((csw&RA)&&(~csw&RB)) stat = CW10;
                if ((csw&RA)&&(csw&RB)) stat = END;
                break;
        }
    }
}

void main() {
    PRTODM2&=~RA&~RB; PRTODM1|=RA|RB; PRTODM0|=RA|RB;
    PRTODR = RA | RB; // RA,RB PullUp
    LCD_1_Start();
    M8C_EnableGInt;

    LCD_1_PrCString("*** PSoC025 ***");
    while(1) {
        LCD_1_Position(1,0);
        LCD_1_PrHexByte(count);
        INT_MSK0 |= INT_MSK0_GPIO;
    }
}
```