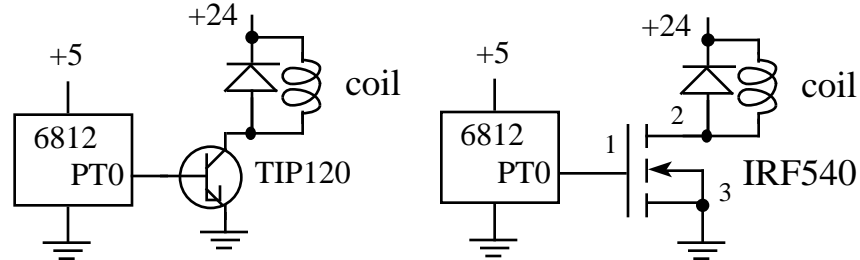


Jonathan W. Valvano
February 28, 2001

(20) Question 1. The 0.5A can be controlled with a TIP120, or a MOSFET IRF540. A ULN2074 could also be used, giving an effective circuit almost identical to the TIP120.



(20) Question 2. A debugging instrument has been added.

(10) Part a) Intrusiveness is the amount of which this instrument affects the system it is trying to test. Typically we need to know the execution speed of this debugging instrument compared to the time required to perform the task we are measuring.

(10) Part b) It will be intrusive if the TxFifo becomes full, because if full OutChar () will spin. On the other hand, if the TxFifo never becomes full, then the execution time for the debugging instrument will be short and bounded.

(60) Question 3. The objective of this program is to design a speed-meter

```
volatile unsigned short Dt; // delay from first to second rising edge in ms
volatile unsigned short Speed; // car speed in mph
// pulse width resolution = 1ms
// speed resolution varies from 1mph at slow speeds to 3mph at high speeds (100mph)
// pulse width range = 34ms to 34.09sec
// speed range = 1mph to 100mph
// IC1, IC0 interrupt each rising edge
void Ritual(void){
    asm(" sei"); // make atomic
    TIOS |= 0x20; // enable OC5
    TSCR |= 0x80; // enable
    TMSK2 = 0x32; // 500 ns clock
    TC5 = TCNT+100; // First one in 50 us
    TIOS &= ~0x03; // PT0, PT1 are both input capture
    DDRT &= ~0x03; // PT0, PT1 are both inputs
    TCTL4 = (TCTL4&0xF0)|0x05; // rising on both PT1 and PT0
    TFLG1 = 0x23; // Clear C5F, C1F, C0F
    TMSK1 |= 0x23; // Arm IC1, IC0, OC5
    asm(" cli"); // enable all interrupts
}
#pragma interrupt_handler TOC5handler()
void TC5handler(void){
    TFLG1 = 0x20; // ack C5F
    TC5 = TC5+2000; // Executed every 1 ms
    if(Dt < 3409){ // prevents overflow
        Dt++; // measures the time from one rising edge to the next in ms
    }
}
#pragma abs_address: ffe4
void (*0Cinterrupt_vector[])() = {
    TC5handler /* ffe4 TC5 */
}
#pragma end_abs_address
```

```
#pragma interrupt_handler IChandler()
void IChandler(void){ // same handler for both IC1, IC0
    if((PORTT&0x03) == 0x03){ // both PT1=PT0=1 on second edge
        if(Dt >=34){ // speed should be less than 100 mph
            Speed = 3409/Dt; // calculate in mph
        }
    }
    else{ // first edge
        Dt = 0; // start time delay measurement
    }
    TFLG1=0x03; // ack both C1F, C0F
}
#pragma abs_address:0xffec
void (*ICinterrupt_vectors[])() = {
    IChandler, /* ffec TC1 */
    IChandler /* ffee TC0 */
}
#pragma end_abs_address
```

Bonus question: The Speed resolution speed resolution = $3409/34 - 3409/35 = 3$ mph