(50) Question 1. In the first implementation, you will use the input capture features.

Part a) To reduce the measurement time we use the smallest possible period measurement resolution of 125ns. With a range of 0 to 10KΩ and a precision of 4096, the desired resistance resolution is 10,000/4096= 2.4414 Ω. The TLC555 converts resistance into period. The clock frequency of a 555 timer is determined by the resistor and capacitor values. The period of a 555 timer is

$$P= 0.693 \bullet C_T \bullet (R_A+2R_B).$$

If we set R_A equal to Rx (Ry), the relationship between period resolution (Δp) and resistance resolution (ΔR) is

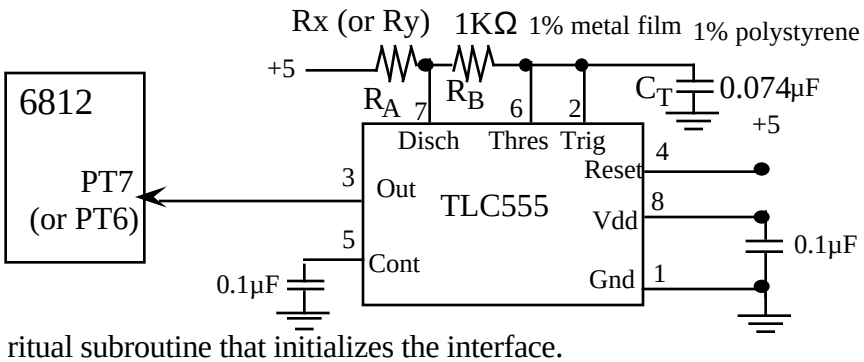
$$\Delta p=0.693 \bullet C_T \bullet \Delta R$$

Thus

$$C_T = \Delta p/(0.693 \bullet \Delta R) = (0.125 \bullet 10^{-6})/(0.693 \bullet 2.4414) = 0.074\ \mu F$$

If we let R_B be equal 1KΩ, then the linear operation is shown in the following table

Rx(Ω)	P (µs)	P (counts)	X=P-819
0	102	819	0
5K	358	2867	2048
10K	614	4915	4096



Part b) Write the ritual subroutine that initializes the interface.

```
// PT7 and PT6 are input = external signals from TLC555
// rising edge to rising edge, resolution = 125ns, Range = 102 µs to 614 us,
// no overflow checking, IC7 and IC6 interrupt each period,
unsigned int Firstx,Firsty;   // TCNT first edges
void Ritual(void){
    asm(" sei");    // make atomic
    TIOS &= 0x3F;   // PT7, PT6 are input capture
    DDRT &= 0x3F;   // PT7, PT6 are input
    TSCR = 0x80;    // enable TCNT
    TMSK2= 0x30;    // 125ns clock
    TCTL3 = (TCTL3&0x0F)|0x50; // rising PT6, PT7
    Firstx=Firsty = TCNT;   // first will be wrong
    TFLG1 = 0xC0;   // Clear C6F C7F
    TMSK1 |= 0xC0;  // Arm C6F C7F
    asm(" cli");}
```

Part c) Write the interrupt service routine

```
#pragma interrupt_handler TC7handler()
void TC7handler(void){unsigned short P=TC7-Firstx; // Rx measurement
    if(P>819)
       X=P-819;
    else
       X=0;       // underflow
    Firstx=TC7;   // Setup for next
    TFLG1=0x80;}  // ack by clearing C7F
#pragma interrupt_handler TC6handler()
void TC6handler(void){unsigned short P=TC6-Firsty; // Ry measurement
    if(P>819)
       Y=P-819;
    else
       Y=0;       // underflow
    Firsty=TC6;   // Setup for next
    TFLG1=0x40;}  // ack by clearing C6F
```

(50) Question 2. In the second implementation, you will use the A/D system.
Part a) We wish to convert R=0 into V=0, and R=10K into V=5 volts. The simple resistor divider will perform the conversion in a nonlinear fashion. Since the sampling frequency is 1 KHz, we add a 500 Hz LPF.

1) select the cutoff frequency, f_c = 500 Hz

2) divide the two capacitors by $2\pi f_c$ (let C_{1A}, C_{2A} be the new capacitor values)

C_{1A} = 141.4µF/$2\pi f_c$= 0.045µF

C_{2A} = 70.7µF/$2\pi f_c$= 0.0225µF

3) locate two standard value capacitors (with the 2/1 ratio) with the same order of magnitude as the desired values

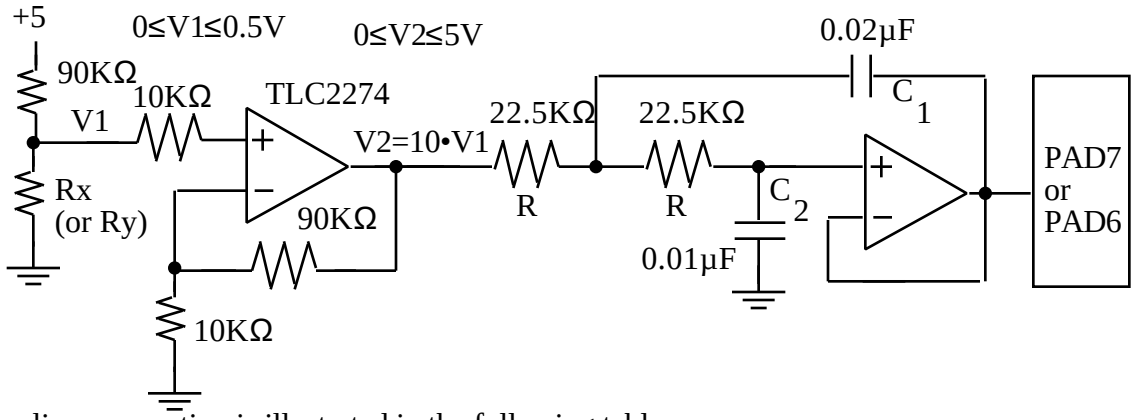
let C_{1B}, C_{2B} be these standard value capacitors, let x be this convenience factor

C_{1B} = C_{1A}/x= 0.02 µF x=0.045/0.02= 2.25

C_{2B} = C_{2A}/x= 0.01 µF

4) adjust the resistors to maintain the cutoff frequency

R = 10KΩ•x = 22.5KΩ. We could have used 20 KΩ resistors and making it a 563 Hz LPF.



The nonlinear operation is illustrated in the following table

Rx(Ω)	V1 (V)	V2 (V)	ADR	X=8*ADR
0	0	0	0	0
5K	0.263	2.63	135	1078
10K	0.5	5	255	2040

Part b) Write the ritual subroutine that initializes the interface.

```
void ritual(void) {
asm(" sei");       // make atomic
  ATDCTL2 = 0x80;  // Activate A/D
  TIOS|=0x01;      // enable OC0
  TSCR|=0x80;      // enable
  TMSK2=0x30;      // 125 ns clock
  TMSK1|=0x01;     // Arm output compare 0
  TFLG1=0x01;      // Initially clear C0F
  TC0=TCNT+8000;   // First one in 1 ms
asm(" cli"); }
```

Part c) Write the output compare channel 0 interrupt service routine

```
#define SCF 0x8000
unsigned short A2D(unsigned char chan){
    ATDCTL5=0x40+chan;    // Start A/D, 8 samples
    while ((ATDSTAT & SCF) == 0){};
    return(ADR0H+ADR1H+ADR2H+ADR3H+ADR4H+ADR5H+ADR6H+ADR7H); } // range is 0 to 1040
#pragma interrupt_handler TC0handler()
void TC0handler(void){
   TC0=TC0+8000;    // Executed every 1 ms
   TFLG1=0x01;      // ack OC5F
   X=A2D(7);
   Y=A2D(6);}
```