

1.Overall Objective

Equipment:

1. Five LEDs
2. Five 220 ohms resistors
3. Five push buttons
4. One 100- nF capacitor

Concept:

The car has 4 door and 1 chasis.

First part:

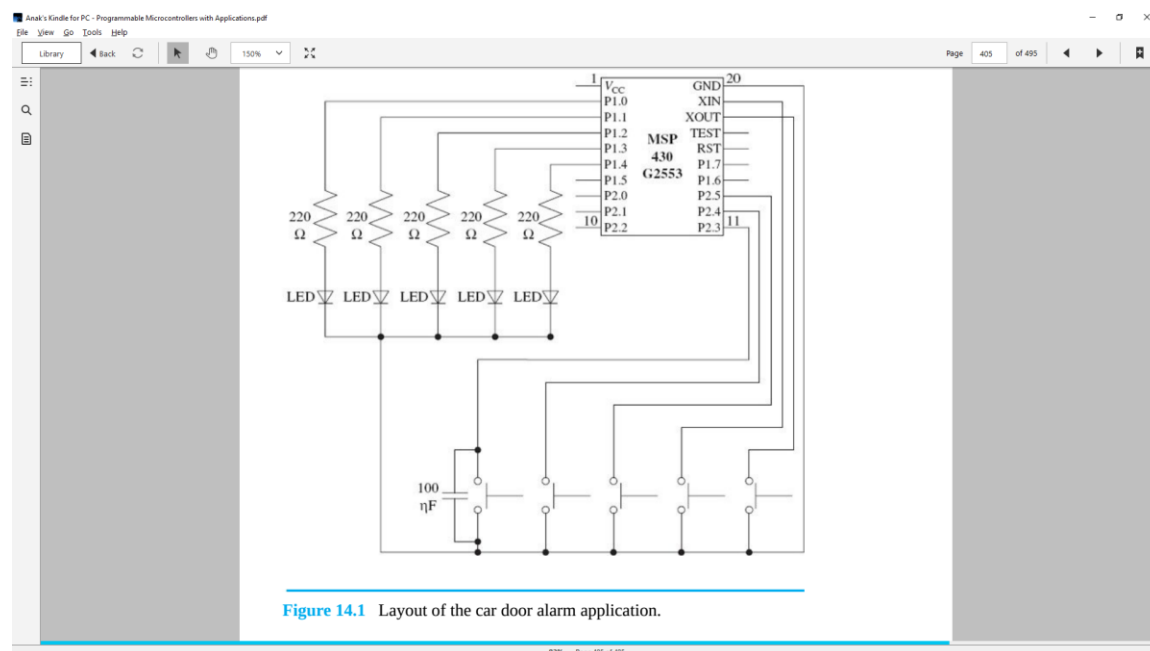
Each door (LED ; each LED has its own button) will be alerted when the door is opened (LED is on). Warning will be reset when the button is pressed (LED turns off).

Second part:

A lock button and lock warning LED is added. When the lock button is pressed, the car is locked and the lock warning LED is off. When the lock button is pressed again, it means that the car is unlocked now which will trigger lock warning to be on.

The lock button cannot be used if all doors are not closed. And as long as the car is locked, doors cannot be opened. To unlock the door, lock button must be pressed again.

2.schematic



3.code

Pseudocode:

First Part:

```
If( door button is pressed) { POUT ^= door LED; // turns door LED on/off }
```

Second Part:

```
If(lock button is pressed & all LED button is off) { POUT ^= lock LED;// turns lock LED on; }
```

```
If (lock button is off) { if(door button is pressed){ POUT ^= door LED; } }
```

Working code:

```
#include <msp430.h>
```

```
int main(void){
```

```
WDTCTL = WDTPW | WDTHOLD
```

```
P1OUT = 0x00;
```

```
P2DIT = 0X00;
```

```
P2REN = 0X1F;
```

```
for(;;){
```

```
If( (P2IN & BIT 4) != BIT4 && (P1OUT & BIT 0) != BIT0 && (P1OUT & BIT 1) != BIT1 && (P1OUT & BIT 2) != BIT2 && (P1OUT & BIT 3) != BIT3)
```

```
{P1OUT ^= BIT4; __delay_cycles(20000); }
```

```
    If(( P1OUT & BIT 4) != BIT4)
```

```
    {
```

```
        if( (BIT0 & P2IN) != BIT0){ P1OUT ^= BIT0; __delay_cycles(20000);}
```

```
        if( (BIT1 & P2IN) != BIT1){ P1OUT ^= BIT1; __delay_cycles(20000);}
```

```
        if( (BIT2 & P2IN) != BIT2){ P1OUT ^= BIT2; __delay_cycles(20000);}
```

```
        if( (BIT3 & P2IN) != BIT3){ P1OUT ^= BIT3; __delay_cycles(20000);}
```

```
    }
```

```
} return 0 ;
```

```
}
```

4.challenges

Hardware:

The hardware of this project is very simple, so I did not encounter any challenges in setting up.

Software:

The difficulty I had with software is whether to select conditions in terms of PxIN or PxOUT. In this case, I must recognize that the lock button is determined by the state of the each door which is represented by P1OUT not P2IN.

Difficulty when converting pseudocode to code:

The line of pseudocode that confused me the most is

```
If(lock button is pressed & all LEDs are off) { POUT ^= lock LED.// turns lock LED on; }
```

This is because I had troubles distinguish “all LEDs are off” from “all LEDs are off **at the same time**” in code language. As a result, I tried using

```
“( P1OUT & (BIT0 | BIT1 | BIT2 | BIT3) ) != (BIT0 | BIT1 | BIT2 | BIT3) ”
```

instead of using

```
“(P1OUT & BIT 0) != BIT0 && (P1OUT & BIT 1) != BIT1 && (P1OUT & BIT 2) != BIT2 && (P1OUT & BIT 3) != BIT3”
```