KAĞAN CENAN

SİDE YILMAZ

LAB #2 REPORT

Question 1)

For the first question, we defined two subroutines which are "DELAY" and "LR".

<u>DELAY SUBROUTINE</u>

```
DELAY:
        cpi R16,2
        brne RR
        ldi R20,20
        rjmp THERE

        RR:     cpi R16,4
                brne RRR
                ldi R20,10
                rjmp THERE

        RRR:ldi R20,1

        THERE:  L1S: ldi R21,50
                L2S: ldi R22,100
                L3S:
                ldi R23,0
                add R23,R16
                in R16, PINB
                cpi R16,0
                brne LOOP
                add R16,R23

                dec R22
                brne L3S
                dec R21
                brne L2S
                dec R20
                brne L1S
        ret
```

In delay subroutine we used nested loops to increase delay time. Here we use R20,R21 and R22 as loop counters. To get different frequencies for different buttons, we change the value of R20 depend on the valu of input register(R16).

So	when the R16=2($01), we use R20=20 (slowest one)

when the R16=4($01), we use R20=10 (medium one)
when the R16=8($01), we use R20=1   (fastest one)

Also, on each the innermost loop, we check if there is a new input or not. It provide to have more accurate data.

<u>LR SUBROUTINE</u>

```
LR:
        ldi R17,$01
        ldi R19,$08
        LEFTS:
                out     PORTD, R17
                RCALL DELAY
                Lsl R17
                dec R19
                brne LEFTS

                ldi R19,$06
                ldi R17,$40

        RIGHTS:
                out     PORTD, R17
                RCALL DELAY
                lsr R17
                dec R19
                brne RIGHTS

                ldi R18,$00
                add R18,R16
RET
```

This subroutine is provide knight rider. We wrote a code that goes to left and then right for one time. At the begining, R17 is equal to 1 and it is the output light. Then we made a loop to shifht left R17 for 8 times.Then when it reach the corner position, R17 started to go throug backward. So the output lights first go throug left and when it reach the left corner, it started to move on right direction. Also, we use delay function before shifting left and shifting right, to observe different frequencie pattens.

After creating thesesub routines, we create 4 cases to represent each button action (i.e. when pushing button 0, it is correcponding to CASE_0). For each button, we write their activities. Since we had already created LR soubroutines, we just put that soubroutine inside to cases. However in LR subroutines, knight rider runs just one time. So, to countinue it’s movement we use the R18 register to hold current input on it. R18 is the history register for our programe. Because, once R16 have a result, when we come back to the begining of the LOOP, R16 become 0 again, since there is no input. But R18 is store te previous value of R16. Hence, if there is no current input on R16 but there was an input from previouse loop, R18 provide to continue it's pattern until there is an input occur.

Question 2)

For the second question, we defined two subroutines which are "DELAY" and "LR" like the first question. Different than the first question we defined "DELAY" differently:

```
DELAY:
                cpi R16,2
                brne C1
                ldi R20,20
                rjmp DELAYPART

        C1:     cpi R16,3
                brne C2
                ldi R20,10
                rjmp DELAYPART

                C2:ldi R20,1

        DELAYPART:
                L1: ldi R21,50
                L2: ldi R22,100
                L3: ldi R23,0
                        add R23,R16
                        in      R16, PINB         ; Read buttons from port D (active low)
                        cpi R16,0
                        brne LOOP
                        add R16,R23

                        dec R22
                        brne L3

                        dec R21
                        brne L2

                        dec R20
                        brne L1
                ret
```

In the code we had a branch which let us to check whether delay needs to be long medium or slow. Firstly we check R16 which comes from the top of the code and tell us its needs to be how long will be the delay. The more important thing is in the DELAYPART is the checking PINB which let us to know button clicked or not. If button is clicked brne LOOP tell us to go back to LOOP which show what should change in the board. Like the first question register R16 decide up to where it comes from as the following and then we ldi to R20 one by one:

R20=20 (slowest one)
R20=10 (medium one)
R20=1 (fastest one)

In the loop part we were storing how many times button clicked in R16.
If the button is not clicked in PORTB :

cpi R16,0;
breq CONT;

let us to continue to loop which comes from past R24 value. As I mentioned R24 store how many clicked happened on the board.

If button is cliked R16 becomes $01 and code goes to CONT branch. In the CONT part we increment R24 value, so we could see how many times button clicked. If the button clicked 5th times we decremented R24 to 1 . In the OK part we did similar thing with the first question. We just checked the R16 value which is already moved from R24. Depends on the click time the Cases happened like the first question. Cases are the same with the first question except the value of R16.

Question 3)

In the 3th question only difference in the code is the subroutine "CLICKDELAY". In the CLICKDELAY we are checking that after clicking is there any click in small time. So we are checking it in CLICKDELAY subroutine.

```
in     R16, PINB              ; Read buttons from port D (active low)
cpi R16,1
breq XX
RJMP YY
XX:    ret

YY:
       dec R22
       brne L3C
       dec R21
       brne L2C

       dec R20
       brne L1C
```

Here we are checking again clicking and then If there is a click code goes back to where it comes from. If there is no click in CLICKDELAY we see that it continue to what is happening.

```
in     R16, PINB
cpi R16,1
breq XX

XX:    ret
```

Provide us to go back.

The subroutines which we used in the q3 are the same with the q2.

In the upper part of the code we used

```
CONT:
              rcall INTER          ;TO GO CL?CKDELAY
              inc R24
              cpi R24,5            ;r24 tickimiz
              brne OK
              ldi R24,1
```

So it lets us the go to CLICKDELAY and if there is a click again in PORTD, with the code ret, it continue to code after rcall. It provides us to increment R24 which again counter of the cases. With the incrementing "OK" branch, program can choose which case must be coming following. Cases are the same with the 2nd question.