EE447 EXPERIMENT #3

PRELIMINARY REPORT

Question 1-) __main.s

```
GPIO_PORTB_DATA     EQU 0x400053FC
                AREA        main, READONLY, CODE
                THUMB
                EXTERN      OutStr  ; Reference external subroutine
                EXTERN      InChar; Serial input Added
                EXTERN      SysTick_Handler;    GPIO signal send
                EXTERN      InitGPIO; GPIO initialize
                IMPORT      Init_Timer
                EXPORT      __main  ; Make available

__main
                BL      InitGPIO; GPIO initialized
                BL      Init_Timer
                MOV     R4, #0x80

Begin           AND     R1, R4, #0x108
                CMP     R1, #0x8
                BEQ     REVERSE
                CMP     R1, #0X100
                BNE     Begin
                MOV     R4, #0x10
                B       Begin
REVERSE         MOV     R4, #0x80
                B       Begin
                ALIGN
                END
```

Init_Timer.s

```
SYSCTRL     EQU 0xE000E010
            AREA rutins , CODE, READONLY
            THUMB
            EXPORT Init_Timer ;

Init_Timer  PROC
            LDR     R0, =SYSCTRL
            MOV     R1, #0
            STR     R1, [R0]
            LDR     R1, =600000
            STR     R1, [R0, #4]
            STR     R1, [R0, #8]
            MOV     R1, #3
            STR     R1, [R0]
            BX      LR

            ALIGN
            ENDP
            END
```

Signalsend.s

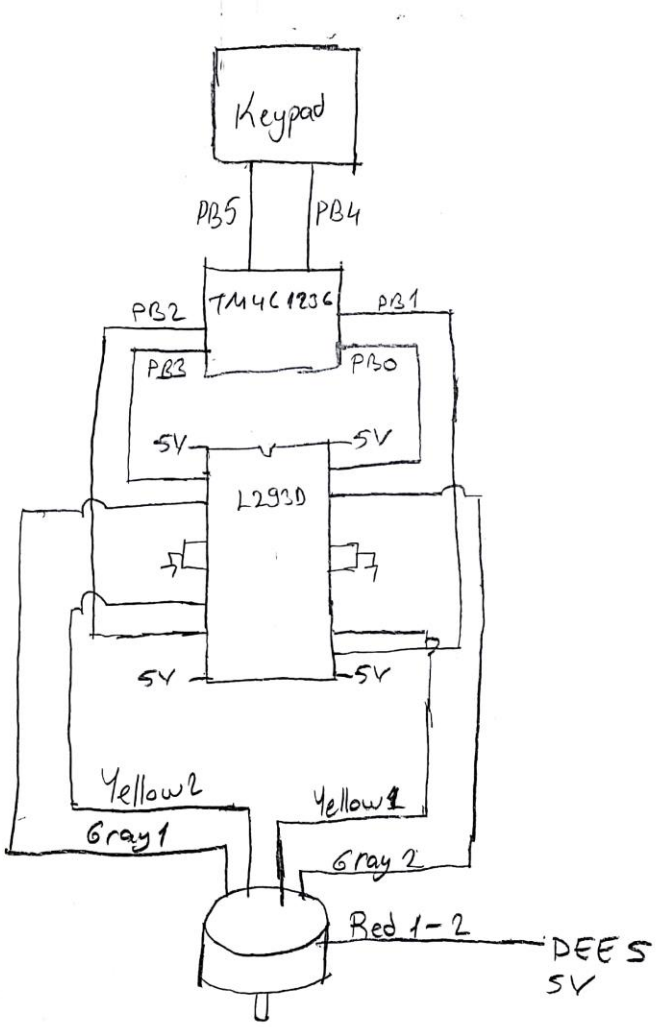
```
GPIO_PORTB_DATA     EQU 0x400053FC

            AREA rutins , CODE, READONLY
            THUMB
            EXPORT SysTick_Handler ;
SysTick_Handler PROC
            LDR     R1,=GPIO_PORTB_DATA; Data address in R1
            MOV     R0, R4
            STR     R0,[R1];    Corresponding Outputs set high
            LSL     R4, R4, #1
            BX LR; end

            ALIGN
            ENDP
            END
```

If we can change LSL with LSR, motor driving direction can be changed.

GPIO_Init.s

```
GPIO_PORTB_DATA     EQU 0x400053FC
GPIO_PORTB_DIR      EQU 0x40005400
GPIO_PORTB_AFSEL    EQU 0x40005420
GPIO_PORTB_DEN      EQU 0x4000551C
GPIO_PORTB_PUR      EQU 0x40005510
GPIO_PORTB_PDR      EQU 0x40005514
IOB                 EQU 0xF0
SYSCTL_RCGCGPIO     EQU 0x400FE608
            AREA rutins , CODE, READONLY
            THUMB
            EXPORT InitGPIO ;
InitGPIO    PROC
            LDR R1 , =SYSCTL_RCGCGPIO
            LDR R0 , [ R1 ]
            ORR R0 , R0 , #0x2;Port B clock enabled
            STR R0 , [ R1 ]
            NOP     ;Wait for clock to stabilize
            NOP
            NOP
            LDR R1 , =GPIO_PORTB_DIR ; Config of Port B starts
            LDR R0 , [ R1 ]
            BIC R0 , #0xFF
            ORR R0 , #IOB;00001111 1->output
            STR R0 , [ R1 ]
            LDR R1 , =GPIO_PORTB_AFSEL
            LDR R0 , [ R1 ]
            BIC R0 , #0xFF
            STR R0 , [ R1 ]
            LDR R1 , =GPIO_PORTB_DEN
            LDR R0 , [ R1 ]
            ORR R0 , #0xFF
            STR R0 , [ R1 ]
            LDR R1 , =GPIO_PORTB_PUR
            LDR R0 , [ R1 ]
            ORR R0 , #0x0F
            STR R0 , [ R1 ]
            BX LR; end
            ALIGN
            ENDP
```

Question 2-)



Question 3-) __main.s

```
 3                AREA        main, READONLY, CODE
 4                THUMB
 5                EXTERN      OutStr  ; Reference external subroutine
 6                EXTERN      InChar; Serial input Added
 7                EXTERN      SignalSend; GPIO signal send
 8                EXTERN      InitGPIO; GPIO initialize
 9                EXTERN      delay; Delay is available
10                EXTERN      CheckInput ; Input Check is available
11                EXPORT      __main  ; Make available
12
13   __main
14                BL      InitGPIO; GPIO initialized
15                MOV     R3, #1;
16                MOV     R6, #0;
17                BL      SignalSend;
18   Begin        BL      CheckInput;
19                CMP     R2, #0
20                BEQ     Begin;
21                CMP     R2, #1
22                BEQ     CW
23                CMP     R2, #2
24                BEQ     CCW
25                B       Begin
26   CW           AND     R3,R3,#15;
27                CMP     R3,#1
28                MOVEQ   R3,#8;
29                LSRNE   R3,R3,#1;
30                MOV     R6,R3;
31                BL      SignalSend;
32                B       Begin;
33   CCW          AND     R3,R3,#15;
34                CMP     R3,#8
35                MOVEQ   R3,#1;
36                LSLNE   R3,R3,#1;
37                MOV     R6,R3;
38                BL      SignalSend;
39                B       Begin;
40                ALIGN
41                END
```

Inputcheck.s

```
1  GPIO_PORTB_DATA     EQU 0x400053FC
2              AREA rutins , CODE, READONLY
3              THUMB
4              EXPORT CheckInput ;
5              EXTERN delay;
6  CheckInput  PROC
7              LDR     R1,=GPIO_PORTB_DATA; Data address in R1
8              MOV     R2,#0; R2 holds the information cw or ccw
9  Check       LDR     R0,[R1];Checks for any input
10             LSR     R0,#4;
11             LSRS    R0,#1;
12             BCC     Delay100
13             LSRS    R0,#1;
14             BCC     Delay100
15             B       Check
16 Delay100    MOV32   R0,#160000;If any input is detected
17             PUSH{LR}
18             BL      delay
19             POP{LR}
20             LDR     R0,[R1];    Check Again
21             LSR     R0,#4;
22             LSRS    R0,#1;
23             MOVCC   R2,#1; 1 is the cw direction PB4 pressed
24             BCC     Released ; If input is detected again wait for relase
25             LSRS    R0,#1;
26             MOVCC   R2,#2; 2 is the ccw direction PB5 pressed
27             BCC     Released
28             BX LR; if no signal
29 Released    LDR     R0,[R1];    It checks for if the switch is open again
30             LSR     R0,#4;
31             LSRS    R0,#1;
32             BCC     Released;   If it is not open
33             LSRS    R0,#1;
34             BCC     Released;
35             BX LR; end
36             ALIGN
37             ENDP
38             END
```

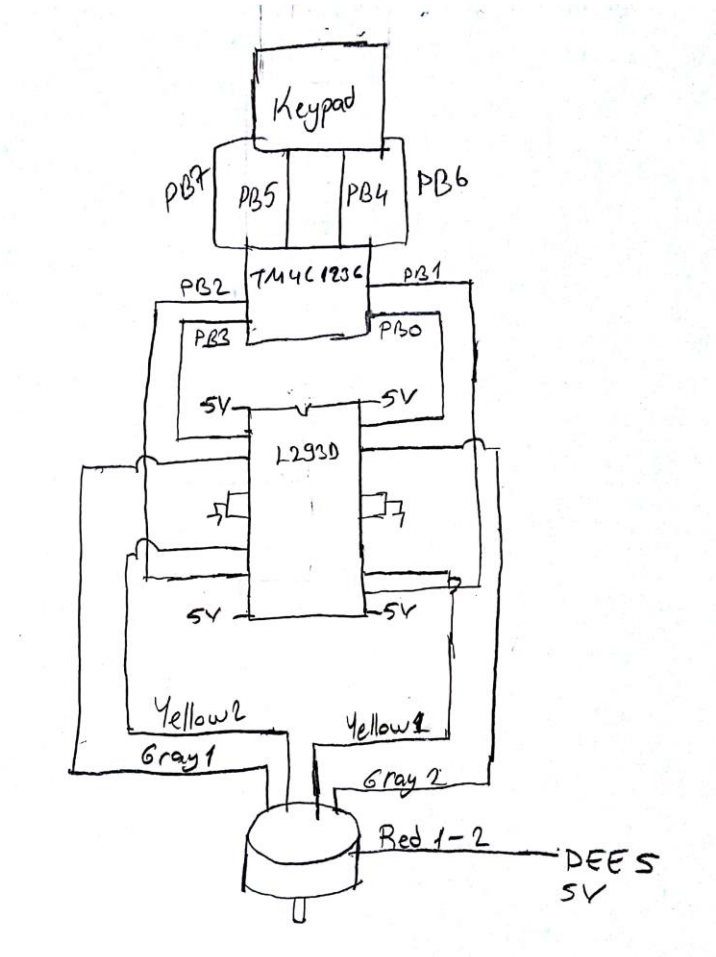
Delay.s

```
1
2              AREA rutins , CODE, READONLY
3              THUMB
4              EXPORT delay ;
5  delay       PROC
6
7  GoBack      SUBS    R0,R0,#1;
8              BEQ     End_Delay
9              B       GoBack
10 End_Delay   BX LR; end
11
12             ALIGN
13             ENDP
14             END
```

Signalsend.s

```
1   GPIO_PORTB_DATA      EQU 0x400053FC
2
3               AREA rutins , CODE, READONLY
4               THUMB
5               EXPORT SignalSend ;
6   SignalSend  PROC
7               LDR      R1,=GPIO_PORTB_DATA; Data address in R1
8               STR      R6,[R1];    Corresponding Outputs set high
9               BX LR; end
10
11              ALIGN
12              ENDP
13              END
```

Question 4-)



Question 5-) In this problem, I specify some modes to determine which buttons are pressed. I write subroutines of 4 events. According to those modes, program runs the subroutine.

__main.s

```
                AREA        main, READONLY, CODE
                THUMB
                EXTERN      SignalSend; GPIO signal send
                EXTERN      InitGPIO; GPIO initialize
                EXTERN      delay; Delay is available
                EXTERN      CHECKINPUT ; Input Check is available
                EXPORT      __main  ; Make available
__main
                BL      InitGPIO; GPIO initialized
                MOV     R3, #1;
                MOV     R6, #0;
                BL      SignalSend;
BEGIN           BL      CHECKINPUT
                B       BEGIN
                ALIGN
                END
```

Checkinput.s

```
GPIO_PORTB_DATA     EQU 0x400053FC
GPIO_PORTB_ICR      EQU 0x4000541C
GPIO_PORTB_RIS      EQU 0x40005414
            AREA rutins , CODE, READONLY
            THUMB
            EXPORT  CHECKINPUT;GPIOPortB_Handler ;
            EXTERN  SignalSend; GPIO signal send
            EXTERN  delay;
CHECKINPUT  PROC
            LDR     R1,=GPIO_PORTB_DATA; Data address in R1
            MOV     R2,#0; R2 holds the information cw or ccw
            MOV     R3, #1;
            MOV32   R7, #1000000
            MOV     R4, #0
Check       MOV     R2, R4
            LDR     R0,[R1];Checks for any input
            LSR     R0,#4;
            LSRS    R0,#1;
            BCC     Delay100
            LSRS    R0,#1;
            BCC     Delay100
            LSRS    R0,#1;
            BCC     Delay100
            LSRS    R0,#1;
            BCC     Delay100
            CMP     R2, #1
            BEQ     CW
            CMP     R2, #2
            BEQ     CCW
            CMP     R2, #3
            BEQ     SPEEDUP
            CMP     R2, #4
            BEQ     SPEEDDOWN
            B       Check
```

```
35  Delay100    MOV32   R0,#160000;If any input is detected
36              PUSH{LR}
37              BL      delay
38              POP{LR}
39              LDR     R0,[R1];    Check Again
40              LSR     R0,#4;
41              LSRS    R0,#1;
42              MOVCC   R2,#1; 1 is the cw direction PB4 pressed
43              BCC     Released ; If input is detected again wait for relase
44              LSRS    R0,#1;
45              MOVCC   R2,#2; 2 is the ccw direction PB5 pressed
46              BCC     Released
47              LSRS    R0,#1;
48              MOVCC   R2,#3; 3 is the speed-up direction PB6 pressed
49              BCC     Released ; If input is detected again wait for relase
50              LSRS    R0,#1;
51              MOVCC   R2,#4; 4 is the speed-down direction PB7 pressed
52              BCC     Released
53              BX LR; if no signal
```

```
56  Released    LDR     R0,[R1];    It checks for if the switch is open again
57              LSR     R0,#4;
58              LSRS    R0,#1;
59              BCC     Released;   If it is not open
60              LSRS    R0,#1;
61              BCC     Released;
62              LSRS    R0,#1;
63              BCC     Released;   If it is not open
64              LSRS    R0,#1;
65              BCC     Released;
66              CMP     R2, #1
67              BEQ     CW
68              CMP     R2, #2
69              BEQ     CCW
70              CMP     R2, #3
71              BEQ     SPEEDUP
72              CMP     R2, #4
73              BEQ     SPEEDDOWN
74
75  CW          AND     R3,R3,#15;
76              CMP     R3,#1
77              MOVEQ   R3,#8;
78              LSRNE   R3,R3,#1;
79              MOV     R6,R3;
80              BL      SignalSend
81              MOV     R0, #0
82              ADD     R0, R7
83              PUSH    {LR}
84              BL      delay
85              POP     {LR}
86              MOV     R4, R2
87              B       Check;
```

```
 89  CCW         AND     R3,R3,#15;
 90              CMP     R3,#8
 91              MOVEQ   R3,#1;
 92              LSLNE   R3,R3,#1;
 93              MOV     R6,R3;
 94              BL      SignalSend;
 95              MOV     R0, #0
 96              ADD     R0, R7
 97              PUSH    {LR}
 98              BL      delay
 99              POP     {LR}
100              MOV     R4, R2
101              B       Check;
102  SPEEDUP
103              MOV32   R7, #160000
104              B       Check
105  SPEEDDOWN
106              MOV32   R7, #1600000
107              B       Check
108              BX LR; end
109
110              ALIGN
111              ENDP
112              END
```