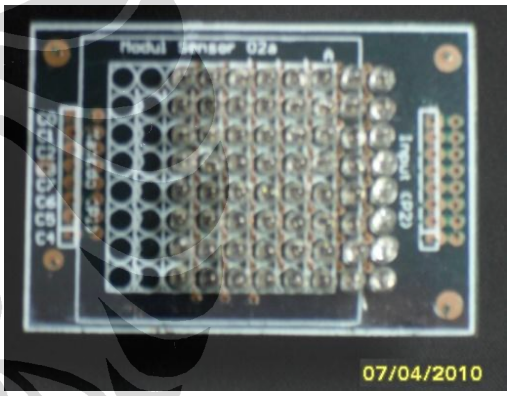
Lampiran 1 : Proses Realisasi.

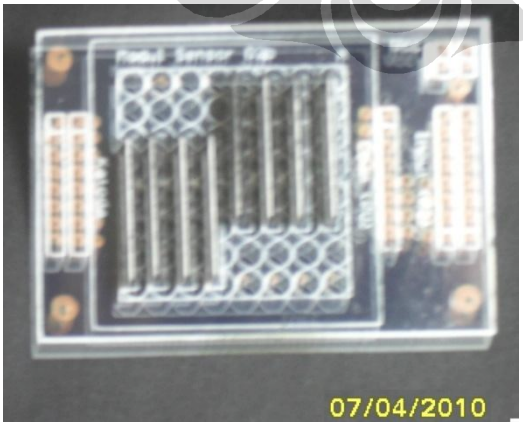
1. Peralatan yang dipakai.

2. Bahan *Cassing*.

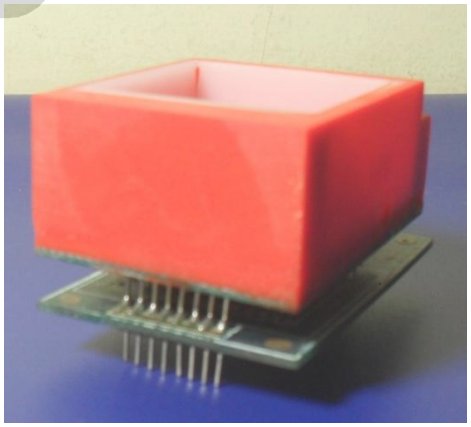
3. Pembuatan modul LED.



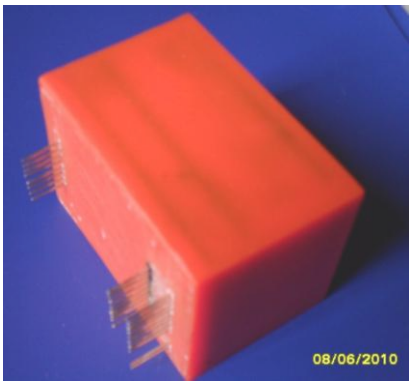
4. Pembuatan modul matrix photodiode bagian atas.



5. Pembuatan modul matrix photodiode bagian bawah.



6. Pembuatan casing.



7. Finishing casing.

8. Pembuatan modul pengambil data.

9. Pembuatan meja uji.

10. Pembuatan modul tampilan data.

11. Pengujian dan pengambilan data.

Lampiran 2 : List program.

Tabel 3.4.1. List main program

```
;=============================================================
;=             MODUL SENSOR MATRIX ARRAY PHOTODIODE          =
;=                 SISTEM SCAN DATA MATRIX ARRAY             =
;=                    UNIVERSITAS INDONESIA                  =
;=                        REV Juni 2010                      =
;=============================================================
; Input     : Receive_code data matrix array ASCII 30h ... 3fh
; Output    : Data posisi kemiringan
; Call Program: 1. Init_LCD.
;               2. T_Motor.
;               3. Program_UTAMA.
;               4. Uji_1.
;-------------------------------------------------------------
            org         0000h
            ljmp        main
;-------------------------------------------------------------
; Inisialisasi peta alamat interup
            org    0003h  ;service routine int ext0 ISR int0
            ljmp   int0isr  ;atau EI0 untuk counter bcd
int0isr     reti
            ;----------------
            org    000bh ;service routine timer0 ISR T0 mode 1
            ljmp   t0isr    ;atau TF0 untuk program jam
t0isr       reti                ;
            ;----------------
            org      0013h  ;service routine intext1 ISR int1
            ljmp     int1isr  ;atau EI1 untuk
int1isr     reti
            ;----------------
            org     001bh    ;service routine timer1 ISR T1
            ljmp    t1isr    ;atau TF1 untuk
t1isr       reti
            ;----------------
            org  0023h ;service routine serial port ISR Tx/Rx
            ljmp        ntx    ;ISR_RX au R1/T1 untuk
ntx         reti
;-------------------------------------------------------------
;           FUNGSI PIN PORT3
;           p3.0  = RxD
;           p3.1  = TxD
;           p3.2  = Int0        ;untuk keyboard
;           p3.3  = Int1        ;untuk printer
;           p3.4  = T0
;           p3.5  = T1
;           p3.7  = I/O
;-------------------------------------------------------------
;           FUNGSI INTERUPT ENABLE
;     bit   simbol      add bit     deskripsi
;     --    -----       ------      ---------
;     IE.7  EA          afh         global enable/disable
;     IE.6  -           aeh         undefined
```

```
;     IE.5  ET2         adh         enable T2 int
;     IE.4  ES          ach         enable serial port int
;     IE.3  ET1         abh         enable T1 int
;     IE.2  EX1         aah         enable external int1
;     IE.1  ET0         a9h         enable T0 int
;     IE.0  EX0         a8h         enable external int0
;-------------------------------------------------------------
            org   0030h
;-------------------------------------------------------------
;           BLOK PROGRAM INISIALISASI
;           1. Stack pointer = 2fh
;           2. Timer0
;           3. Timer1
;           4. Keyboard PC-AT
;           5. LCD
;           6. Printer
;-------------------------------------------------------------
main:
SetupRDY:   mov         r7,#0ffh
a1          mov         r6,#ffh
a2          djnz        r6,a2
            djnz        r7,a1
            ;-----------------
            ;
            mov         sp,#5fh
            clr         p1.0
            clr         p1.1
            clr         p1.2
            clr         p2.7        ;back light LCD
;-------------------------------------------------------------
;Main program demo tampilan message
;-------------------------------------------------------------
          lcall       init_LCD

UJI_1       lcall       t_motor
            lcall       PROGRAM_UTAMA
            sjmp        UJI_1

             end
;=============================================================
```

Tabel 3.4.2. List sub program pengambilan data

```
;=============================================================
;=             MODUL SENSOR MATRIX ARRAY PHOTODIODE          =
;=                 SISTEM SCAN DATA MATRIX ARRAY             =
;=                    UNIVERSITAS INDONESIA                  =
;=                        REV Juni 2010                      =
;=============================================================
; Fungsi    : Mengambil data dari matrix array photodiode
; Input     : Sensor matrix array photodiode 8 x 8
; Output    : Data Hexa 8 byte
; SubProgram: 1. Scan_Code
;             2. Transmit_Code
; Waktu Total : 8987M = 8,987 mS
;-------------------------------------------------------------
            org         0000h
            ljmp        main
;-------------------------------------------------------------
; Inisialisasi peta alamat interrup
            org    0003h    ;service routine int ext0 ISR int0
            clr    ie.0           ;disable int0
            ljmp   Transmit_CODE ;atau EI0 untuk counter bcd
int0isr     reti
            ;----------------
            org    000bh ;service routine timer0 ISR T0 mode 1
            ljmp   t0isr ;atau TF0 untuk program jam
t0isr       reti                  ;
            ;----------------
            org    0013h   ;service routine int ext1 ISR int1
            ljmp   int1isr  ;atau EI1 untuk
int1isr     reti
            ;----------------
            org    001bh   ;service routine timer1 ISR T1
            ljmp   t1isr     ;atau TF1 untuk
t1isr       reti
            ;----------------
            org    0023h  ;service routine serial ISR Tx/Rx
            ljmp        ntx    ; ISR_RX      ;atau R1/T1 untuk
ntx         reti
;-------------------------------------------------------------
;           FUNGSI PIN PORT3
;           p3.0  = RxD
;           p3.1  = TxD
;           p3.2  = Int0      ;untuk keyboard
;           p3.3  = Int1      ;untuk printer
;           p3.4  = T0
;           p3.5  = T1
;           p3.7  = I/O
;-------------------------------------------------------------
;           FUNGSI INTERUPT ENABLE
;     bit   simbol      add bit     deskripsi
;     --    -----       ------      ---------
;     IE.7  EA          afh         global enable/disable
;     IE.6  -           aeh         undefined
;     IE.5  ET2         adh         enable T2 int
;     IE.4  ES          ach         enable serial port int
```

```
;     IE.3  ET1         abh         enable T1 int
;     IE.2  EX1         aah         enable external int1
;     IE.1  ET0         a9h         enable T0 int
;     IE.0  EX0         a8h         enable external int0
;=============================================================
            org   0030h
main:       mov         ie,#00h
SetupRDY:   mov         r7,#100
a1          mov         r6,#200
a2          djnz        r6,a2
            djnz        r7,a1
            mov         2ch,#20
            mov         2dh,#30h
            mov         sp,#5fh
;-------------------------------------------------------------
            ;enable int0
            setb        p3.2
            mov         ie,#81h
            Sjmp        $
;=============================================================
; Input     : 20h ... 2fh
; Output    : serial data mode SPI (serial peripheral
interface
;             p2.0 = output clock '0' ke rx
;             p2.1 = output data ke rx
;                    status pertama sebagai acknowledge int0
;Register   : r0 = alamat data
;             r2 = counter bit
;             r3 = counter byte
;Proses     : 1. Scan_CODE untuk simulasi memakai MODE_SCAN
;             2. Trans_conv
;             3. Proses kirim serial
; Waktu Total : 8987M = 8,987 mS
;=============================================================
ISR_INT0:
Transmit_CODE:
            push        e0h
            push        d0h
            push        03h
            push        02h
            push        00h
            ;--------------
            lcall       Scan_Code
            lcall       TRANS_CONV
            ;proses kirim serial
            mov         r0,#20h
            mov         r3,#08
tx_2        mov         r2,#08
            ;------------------
            mov         a,@r0
tx_1        rlc         a
            ;data dikirim
            mov         p3.2,c
            ;clock diaktifkan '0' dan tunggu '1' dari rx
            clr         p3.1
```

```
            lcall       dly_10us
            lcall       dly_10us
            lcall       dly_10us
            lcall       dly_100us
            setb        p3.1
            ;
            djnz        r2,tx_1
            ;selesai satu byte
            inc         r0
            djnz        r3,tx_2
            ;selesai semua 16 byte
            ;
            pop         00h
            pop         02h
            pop         03h
            pop         d0h
            pop         e0h
            ;--------------
            ;into di enable lagi
            setb        p3.2
            setb        ie.0
            ;
            RETI
           ;--------------------
tx_1a       clr         p3.1
            jnb         p3.1,$
            sjmp        tx_1
tx_2a       mov         r2,#08
            clr         p3.1
            jnb         p3.1,$
            sjmp        tx_1
;================================================================
; Baca data scan Code Matrix Array
; Port_0 : '0' --> R0=p0.0 .... R7=p0.7
; Port_1 : '1' --> C0=p1.0 .... C3=p1.3 ; C4=p1.7 .... C7=p1.4
; Buffer input = p2
; Buffer data = 20h ..... 27h
; sp = 5fh
; Total waktu = 151 M = 151 uS
;================================================================
Scan_Code:  mov         p0,#00h
            mov         p1,#ffh
            mov         r0,#20h
            ;
            push        e0h
            push        d0h
            push        00h
            ;--------------
            clr         p1.0
            lcall       dly_10us
            mov         a,p2
            clr         a.1
            mov         @r0,a
            inc         r0
            setb        p1.0
```

```
;---------------
clr         p1.1
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
setb        p1.1
;---------------
clr         p1.2
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
setb        p1.2
;---------------
clr         p1.7
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
setb        p1.7
;---------------
clr         p1.6
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
setb        p1.6
;---------------
clr         p1.5
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
setb        p1.5
;---------------
clr         p1.4
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
setb        p1.4
;---------------
clr         p1.3
lcall       dly_10us
mov         a,p2
clr         a.1
mov         @r0,a
inc         r0
```

```
            setb        p1.3
            ;---------------
            pop         00h
            pop         d0h
            pop         e0h
            ;
            Ret
            ;
dly_10us:   nop          ;1 M
            nop          ;1M
            nop          ;1M
            nop          ;1M
            nop          ;1M
            nop          ;1M
            ret         ;2M total= 8 M + lcall 2M = 10M = 10 uS
            ;
dly_100us:  mov        r7,#7
dd1         lcall      dly_10us
            djnz       r7,dd1
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            ret          ; total = 100M

            end
;===============================================================
```

Tabel 3.4.3. List sub program receive code.

```
;=============================================================
;             SUB PROGRAM Receive_Code
; Fungsi  : Untuk mengambil data sensor dari modul matrix
array.
; Mode    : Serial peripheral interface (SPI)
; Input   : 20h ... 2fh
; Output  : serial data tanpa RS232
;             p3.5 = output clock serial '0'
;             p3.6 = input-output data serial
;                    status pertama sebagai acknowledge int0
;Register : r0 = alamat data
;             r2 = counter bit
;             r3 = counter byte
;Proses   : 1. Scan_CODE untuk simulasi memakai MODE_SCAN
;           2. Trans_conv
;           3. Proses kirim serial
; Waktu Total : 900 M = 900 µS.
;=============================================================
Receive_CODE:
            push        e0h
            push        d0h
            push        03h
            push        02h
            push        00h
            ;proses terima serial
            ;sinyal start/reques
            clr         p3.6
            nop
            setb        p3.6
            setb        p3.5
            ;
            mov         r0,#20h
            mov         r3,#08
rx_2        mov         r2,#08
            ;-----------------
            ;tunggu clock diaktifkan '0'
rx_1        jb          p3.5,$
            ;
            mov         c,p3.6
            mov         a,@r0
            rlc         a
            ;data diambil
            mov         @r0,a
            ;
            djnz        r2,rx_1a
            ;selesai satu byte
            inc         r0
            djnz        r3,rx_2a
            ;selesai semua 16 byte
            ;
            pop         00h
            pop         02h
            pop         03h
            pop         d0h
```

```
            pop         e0h
            ;--------------
            ;into di enable lagi
            setb        p3.6
            setb        p3.5
            ;
            clr         01h
            clr         09h
            clr         11h
            clr         19h
            clr         21h
            clr         29h
            clr         31h
            clr         39h
            ret
            ;tunggu clock diaktifkan '1'
rx_1a       jnb         p3.5,$
            sjmp        rx_1
rx_2a       mov         r2,#08
            jnb         p3.5,$
            sjmp        rx_1
;=============================================================
```

Tabel 3.4.4. List sub program trans_conv.

```
;=============================================================
;                SUB PROGRAM Trans_Conv
; Fungsi : Untuk mengkonversi bentuk Tabel data Row (20h-27h)
;          ke bentuk Colom (28h-2fh)
; Input  : Alamat 20h – 27h
; Output : Alamat 20h – 27h dan 28hh – 2fh
;        1. Data Row untuk posisi depan & belakang
;        2. Data Colom untuk posisi kanan & kiri
;        3. Lihat keterangan pada subroutine CARI_POSISI
; Waktu Total : 1.456 M = 1.456 µS.
;=============================================================
trans_conv: mov         r0,#20h
            mov         r1,#28h
            mov         r2,#00          ; up
            mov         r3,#08h         ; down
            ;proses transfer konversi Tabel data x-->y
rotet:      push        00h
            inc         r2
            push        02h
            mov         a,@r0
rot1        rlc         a
            djnz        r2,rot1
            MOV         f7h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot2        rlc         a
            djnz        r2,rot2
            MOV         f6h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot3        rlc         a
            djnz        r2,rot3
            MOV         f5h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot4        rlc         a
            djnz        r2,rot4
            MOV         f4h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot5        rlc         a
            djnz        r2,rot5
```

```
            MOV         f3h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot6        rlc         a
            djnz        r2,rot6
            MOV         f2h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot7        rlc         a
            djnz        r2,rot7
            MOV         f1h,c
            pop         02h
            ;
            push        02h
            inc         r0
            mov         a,@r0
rot8        rlc         a
            djnz        r2,rot8
            MOV         f0h,c
            pop         02h
            ;---selasai 1 byte
            mov         a,f0h
            mov         @r1,a
            ;---
            pop         00h
            inc         r1
            djnz        r3,rotet
            ;---selesai semua byte
            Ret
            end
;===============================================================
```

Tabel 3.4.5. List sub program hitung_1.

```
;=============================================================
; Proses hitungan data
; Data: Depan lsb 30h msb 31h
;       Belakang lsb 32h msb 33h
;       Kanan lsb 34h msb 35h
;       Kiri lsb 36h msb 37h
; Data temporer 38h
; Output reg. B = 00 = normal
;               = 01 = depan
;               = 02 = belakang
;               = 03 = kanan
;               = 04 = kiri
; Waktu Total : 2.161 M = 2.161 µS.
;=============================================================
HITUNG:      mov        r0,#30h
             mov        r1,#08
             clr        a
clr_bf       mov        @r0,a
             inc        r0
             djnz       r1,clr_bf
             ;
             lcall      Depan
             lcall      Belakang
             lcall      Kanan
             lcall      Kiri
             ;---------------
             ; bandingkan semua msb
             mov        a,31h
             cjne       a,#00h,dm_ada
             mov        a,33h
             cjne       a,#00h,dm_ada
             mov        a,35h
             cjne       a,#00h,dm_ada
             mov        a,37h
             cjne       a,#00h,dm_ada
             ; bandingkan semua lsb
             mov        a,30h
             cjne       a,#00h,dm_ada
             mov        a,32h
             cjne       a,#00h,dm_ada
             mov        a,34h
             cjne       a,#00h,dm_ada
             mov        a,36h
             cjne       a,#00h,dm_ada
             ;berarti nol semua
             ;set b = 00h = normal
             ljmp       keluar_hitung
             ; Bandingkan depan - belakang
dm_ada       clr        c
             mov        a,30h
             subb       a,32h
             mov        a,31h
             subb       a,33h
             jc         b_besar
```

```
                ;------------
                ;d_besar
                ; Bandingkan  depan - kanan
                clr         c
                mov         a,30h
                subb        a,34h
                mov         a,31h
                subb        a,35h
                jc          kn_besar
                ;------------
                ;d_besar
                ; Bandingkan  depan - kiri
                clr         c
                mov         a,30h
                subb        a,36h
                mov         a,31h
                subb        a,37h
                jc          kr_besar
                ; depan paling_besar
                mov         b,#01h
                ljmp        keluar_hitung
                ; Bandingkan  belakang - kanan
b_besar         clr         c
                mov         a,32h
                subb        a,34h
                mov         a,33h
                subb        a,35h
                jc          kn_besar
                ;------------
                ;b_besar
                ; Bandingkan  belakang - kiri
                clr         c
                mov         a,32h
                subb        a,36h
                mov         a,33h
                subb        a,37h
                jc          kr_besar
                ;------------
                ; belakang paling_besar
                mov         b,#02h
                ljmp        keluar_hitung
                ; Bandingkan  kanan - kiri
kn_besar        clr         c
                mov         a,34h
                subb        a,36h
                mov         a,35h
                subb        a,37h
                jc          kr_besar
                ;------------
                ; kanan paling_besar
                mov         b,#03h
                ljmp        keluar_hitung
                ;------------
                ; kiri paling_besar
kr_besar        mov         b,#04h
```

```
             ljmp       keluar_hitung
keluar_hitung RET
             sjmp       $
;-------------------------------------------------------------
Depan:       mov        r0,#20h
             lcall      jumlah_d
             ; alamat 20h dikalikan n=1
             ; mul ab --> msb di b lsb di a
             mov        a,38h
             mov        b,#01h
             mul        ab
             lcall      total_d
             ;---------------
             ; alamat 21h
             mov        r0,#21h
             lcall      jumlah_d
             ; alamat 21h dikalikan n=2
             ; mul ab --> msb di b lsb di a
             mov        a,38h
             mov        b,#02h
             mul        ab
             lcall      total_d
             ;---------------
             ; alamat 22h
             mov        r0,#22h
             lcall      jumlah_d
             ; alamat 22h dikalikan n=3
             ; mul ab --> msb di b lsb di a
             mov        a,38h
             mov        b,#03h
             mul        ab
             lcall      total_d
             ;---------------
             ; alamat 23h
             mov        r0,#23h
             lcall      jumlah_d
             ; alamat 23h dikalikan n=4
             ; mul ab --> msb di b lsb di a
             mov        a,38h
             mov        b,#04h
             mul        ab
             lcall      total_d
             ;---------------
             ; alamat 24h
             mov        r0,#24h
             lcall      jumlah_d
             ; alamat 21h dikalikan n=5
             ; mul ab --> msb di b lsb di a
             mov        a,38h
             mov        b,#05h
             mul        ab
             lcall      total_d
             ;---------------
             ; alamat 25h
             mov        r0,#25h
```

```
            lcall       jumlah_d
            ; alamat 25h dikalikan n=6
            ; mul ab --> msb di b lsb di a
            mov         a,38h
            mov         b,#06h
            mul         ab
            lcall       total_d
            ;---------------
            ; alamat 26h
            mov         r0,#26h
            lcall       jumlah_d
            ; alamat 26h dikalikan n=7
            ; mul ab --> msb di b lsb di a
            mov         a,38h
            mov         b,#07h
            mul         ab
            lcall       total_d
            ;---------------
            ; alamat 27h
            mov         r0,#27h
            lcall       jumlah_d
            ; alamat 27h dikalikan n=8
            ; mul ab --> msb di b lsb di a
            mov         a,38h
            mov         b,#08h
            mul         ab
            lcall       total_d
            ;-----depan selesai
            RET
jumlah_d:
            mov         r3,#08      ;counter bit
            mov         38h,#00h
            mov         a,@r0
ht_d1       rlc         a
            jc          tmp_d
ht_d2       djnz        r3,ht_d1
            ret
            ;data temporer
tmp_d       inc         38h
            sjmp        ht_d2
            ;
total_d:    add         a,30h
            mov         30h,a
            mov         a,f0h       ;reg b
            addc        a,31h
            mov         31h,a
            ret
;-------------------------------------------------------------
Belakang:   mov         r0,#27h
            lcall       jumlah_b
            ; alamat 27h dikalikan n=1
            ; mul ab --> msb di b lsb di a
            mov         a,38h
            mov         b,#01h
            mul         ab
```

```
lcall       total_b
;---------------
; alamat 26h
mov         r0,#26h
lcall       jumlah_b
; alamat 26h dikalikan n=2
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#02h
mul         ab
lcall       total_b
;---------------
; alamat 25h
mov         r0,#25h
lcall       jumlah_b
; alamat 25h dikalikan n=3
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#03h
mul         ab
lcall       total_b
;---------------
; alamat 24h
mov         r0,#24h
lcall       jumlah_b
; alamat 24h dikalikan n=4
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#04h
mul         ab
lcall       total_b
;---------------
; alamat 23h
mov         r0,#23h
lcall       jumlah_b
; alamat 23h dikalikan n=5
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#05h
mul         ab
lcall       total_b
;---------------
; alamat 22h
mov         r0,#22h
lcall       jumlah_b
; alamat 22h dikalikan n=6
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#06h
mul         ab
lcall       total_b
;---------------
; alamat 21h
mov         r0,#21h
lcall       jumlah_b
```

```
                ; alamat 21h dikalikan n=7
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#07h
                mul         ab
                lcall       total_b
                ;---------------
                ; alamat 20h
                mov         r0,#20h
                lcall       jumlah_b
                ; alamat 20h dikalikan n=8
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#08h
                mul         ab
                lcall       total_b
                ;-----depan selesai
                RET
jumlah_b:
                mov         r3,#08      ;counter bit
hit_depan       mov         38h,#00h
                mov         a,@r0
ht_b1           rlc         a
                jc          tmp_b
ht_b2           djnz        r3,ht_b1
                ret
                ;data temporer
tmp_b           inc         38h
                sjmp        ht_b2
                ;
total_b:        add         a,32h
                mov         32h,a
                mov         a,f0h       ;reg b
                addc        a,33h
                mov         33h,a
                ret
;-------------------------------------------------------------
Kanan:          mov         r0,#28h
                lcall       jumlah_kn
                ; alamat 28h dikalikan n=1
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#01h
                mul         ab
                lcall       total_kn
                ;---------------
                ; alamat 29h
                mov         r0,#29h
                lcall       jumlah_kn
                ; alamat 29h dikalikan n=2
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#02h
                mul         ab
                lcall       total_kn
```

```
;---------------
; alamat 2ah
mov         r0,#2ah
lcall       jumlah_kn
; alamat 2ah dikalikan n=3
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#03h
mul         ab
lcall       total_kn
;---------------
; alamat 2bh
mov         r0,#2bh
lcall       jumlah_kn
; alamat 2bh dikalikan n=4
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#04h
mul         ab
lcall       total_kn
;---------------
; alamat 2ch
mov         r0,#2ch
lcall       jumlah_kn
; alamat 2ch dikalikan n=5
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#05h
mul         ab
lcall       total_kn
;---------------
; alamat 2dh
mov         r0,#2dh
lcall       jumlah_kn
; alamat 2dh dikalikan n=6
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#06h
mul         ab
lcall       total_kn
;---------------
; alamat 2eh
mov         r0,#2eh
lcall       jumlah_kn
; alamat 2eh dikalikan n=7
; mul ab --> msb di b lsb di a
mov         a,38h
mov         b,#07h
mul         ab
lcall       total_kn
;---------------
; alamat 2fh
mov         r0,#2fh
lcall       jumlah_kn
; alamat 2fh dikalikan n=8
```

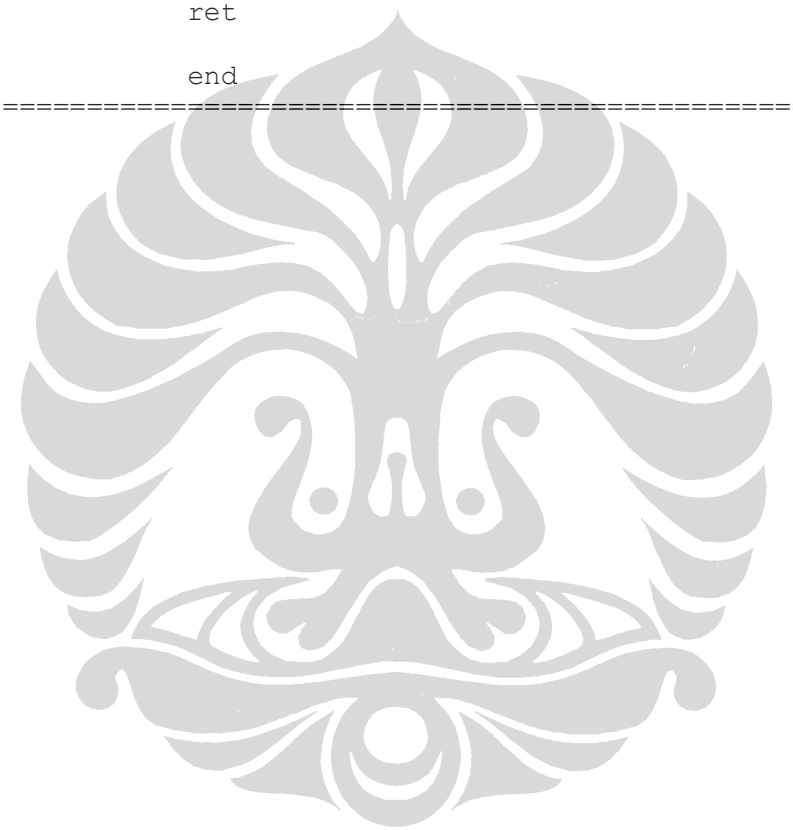
```
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#08h
                mul         ab
                lcall       total_kn
                ;-----depan selesai
                RET

jumlah_kn:
                mov         r3,#08      ;counter bit
hit_kanan       mov         38h,#00h
                mov         a,@r0
ht_kn1          rlc         a
                jc          tmp_kn
ht_kn2          djnz        r3,ht_kn1
                ret
                ;data temporer
tmp_kn          inc         38h
                sjmp        ht_kn2
                ;
total_kn:       add         a,34h
                mov         34h,a
                mov         a,f0h       ;reg b
                addc        a,35h
                mov         35h,a
                ret
;-------------------------------------------------------------
Kiri:           mov         r0,#2fh
                lcall       jumlah_kr
                ; alamat 2fh dikalikan n=1
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#01h
                mul         ab
                lcall       total_kr
                ;---------------
                ; alamat 2eh
                mov         r0,#2eh
                lcall       jumlah_kr
                ; alamat 2eh dikalikan n=2
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#02h
                mul         ab
                lcall       total_kr
                ;---------------
                ; alamat 2dh
                mov         r0,#2dh
                lcall       jumlah_kr
                ; alamat 2dh dikalikan n=3
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#03h
                mul         ab
                lcall       total_kr
```

```
                ;---------------
                ; alamat 2ch
                mov         r0,#2ch
                lcall       jumlah_kr
                ; alamat 2ch dikalikan n=4
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#04h
                mul         ab
                lcall       total_kr
                ;---------------
                ; alamat 2bh
                mov         r0,#2bh
                lcall       jumlah_kr
                ; alamat 2bh dikalikan n=5
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#05h
                mul         ab
                lcall       total_kr
                ;---------------
                ; alamat 2ah
                mov         r0,#2ah
                lcall       jumlah_kr
                ; alamat 2ah dikalikan n=6
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#06h
                mul         ab
                lcall       total_kr
                ;---------------
                ; alamat 29h
                mov         r0,#29h
                lcall       jumlah_kr
                ; alamat 29h dikalikan n=7
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#07h
                mul         ab
                lcall       total_kr
                ;---------------
                ; alamat 28h
                mov         r0,#28h
                lcall       jumlah_kr
                ; alamat 28h dikalikan n=8
                ; mul ab --> msb di b lsb di a
                mov         a,38h
                mov         b,#08h
                mul         ab
                lcall       total_kr
                ;-----depan selesai
                RET
jumlah_kr:
                mov         r3,#08      ;counter bit
hit_kiri        mov         38h,#00h
```

```
            mov         a,@r0
ht_kr1      rlc         a
            jc          tmp_kr
ht_kr2      djnz        r3,ht_kr1
            ret
            ;data temporer
tmp_kr      inc         38h
            sjmp        ht_kr2
            ;
total_kr:   add         a,36h
            mov         36h,a
            mov         a,f0h       ;reg b
            addc        a,37h
            mov         37h,a
            ret

            end
;===============================================================
```



Tabel 3.4.6.List sub program cari_posisi.

```
;=============================================================
;               SUB PROGRAM Cari_Posisi
; Fungsi    : Untuk menentukan arah posisi Kemiringan
; Prinsip   : Menjumlahkan dgn bobot yg berbeda antara arah F.
; Input     : 20h ... 2fh
; Output    : register B
; Untuk mencari posisi kemiringan jika :
;           1. Depan dimulai dari alamat 20h=ffh --> 27h=ffh
;           2. Belakang dimulai dari almat 2fh=ffh --> 20h=ffh
;           3. Kanan dimulai dari alamat 28h=ffh --> 2fh=ffh
;           4. Kiri dimulai dari alamat 2fh=ffh --> 27h=ffh
; Register: r0 = address input
;           7eh = counter derajat kemiringan atau posisi
;                 01 = posisi_1
;                 02 = posisi_2
;                 ............
;                 08 = posisi_8
;           r2 = counter address
;           b  = flag arah kemiringan
;                - flag_depan    b.0 = '1'
;                - flag_belakang b.1 = '1'
;                - flag_kanan    b.2 = '1'
;                - flag_kiri     b.3 = '1'
; Waktu Total : ……. M= …….. mS
;=============================================================
CARI_POSISI:
            push        e0h
            push        d0h
            push        02h
            push        00h
            ;--------------
            ;dimulai dari arah depan
            ;semua flag posisi  dihapus dan counter posisi
            clr          a
            mov           b,a
            mov           7eh,a
            ;---------------
            ;cek depan
depan       mov          r0,#20h     ;up address
            mov          7eh,#00h
            mov          r2,#08
dpn_1       mov          a,@r0
            cjne         a,#ffh,cek_dpn
            setb         b.0                    ;flag_depan
            inc          7eh
            inc          r0
            djnz         r2,dpn_1
            ;
cek_dpn     jnb          b.0,belakang
            mov          a,@r0
            cjne         a,#24h,dpn_3
            inc          r0
            inc          7eh
            djnz         r2,dpn_1
```

```
            ;
dpn_3       mov         a,r2
            cjne        a,#00h,dpn_2
            sjmp        ada_posisi

dpn_2       inc         r0
            dec         r2
            sjmp        dpn_1
            ;---------------
            ;cek belakang
belakang    mov         r0,#27h     ;down address
            mov         7eh,#00h
            mov         r2,#08
blk_1       mov         a,@r0
            cjne        a,#ffh,cek_blk
            setb        b.1                     ;flag_belakang
            inc         7eh
            dec         r0
            djnz        r2,blk_1
            ;
cek_blk     jnb         b.1,kanan
            mov         a,r2
            cjne        a,#00h,blk_2
            sjmp        ada_posisi
blk_2       dec         r0
            dec         r2
            sjmp        blk_1
            ;---------------
            ;cek kanan
kanan       mov         r0,#28h     ;up address
            mov         7eh,#00h
            mov         r2,#08
kkn_1       mov         a,@r0
            cjne        a,#ffh,cek_kkn
            setb        b.2                     ;flag_kanan
            inc         7eh
            inc         r0
            djnz        r2,kkn_1
            ;
cek_kkn     jnb         b.2,kiri
            mov         a,r2
            cjne        a,#00h,kkn_2
            sjmp        ada_posisi
kkn_2       inc         r0
            dec         r2
            sjmp        kkn_1
            ;cek kiri
kiri        mov         r0,#2fh     ;down address
            mov         7eh,#00h
            mov         r2,#08
kri_1       mov         a,@r0
            cjne        a,#ffh,cek_kri
            setb        b.3
            inc         7eh
            dec         r0
```

```
            djnz        r2,kri_1
            ;
cek_kri     jnb         b.3,out_cari_posisi
            mov         a,r2
            cjne        a,#00h,kri_2
            sjmp        ada_posisi
kri_2       dec         r0
            dec         r2
            sjmp        kri_1
kri_1a      mov         a,@r0
            cjne        a,#ffh,cek_kri
            setb        b.3
            inc         7eh
            dec         r0
            djnz        r2,kri_1a
            ;
;cek_kri    jnb         b.3,out_cari_posisi
            sjmp        ada_posisi
            ;--- selesai scan posisi
out_cari_posisi
ada_posisi
                pop     00h
                pop     02h
                pop     d0h
                pop     e0h
                ret

               end
;=============================================================
```

Tabel 3.4.7. List sub program hitung_2.

```
;=============================================================
;              SUB PROGRAM HITUNG_2
; Proses hitungan data tanpa bobot kemudian semua dikalikan 4
; data depan = data belakang di 39h
; data kanan = data kiri di 3ah
; Waktu Total : 789 M = 789 µS.
;=============================================================
HITUNG2:
            mov         a,#00h
            mov         r0,#39h
            mov         r2,#07h
clr_x       mov         @r0,a
            inc         r0
            djnz        r2,clr_x
            lcall       data_depan
            lcall       data_kanan
            ret
            ;----------
data_depan:
            mov         r0,#20h
            mov         r2,#08h
dat_d0      mov         r3,#08h
            mov         a,@r0
dat_d1      rlc         a
            jc          data_d
dat_d2      djnz        r3,dat_d1
            inc         r0
            djnz        r2,dat_d0
            ;semua dikalikan 4
            ;disimpan 3ch msb  3dh lsb
            mov         a,39h
            cjne        a,#01h,x1
            mov         a,#00h
            mov         39h,a
            mov         f0h,a
x1          add         a,e0h
            da          a
            mov         3ch,a
            clr         a
            addc        a,#00h
            da          a
            mov         3dh,a
            ret
            ;data temporer
data_d      push        e0h
            mov         a,39h
            add         a,#01h
            da          a
            mov         39h,a
            pop         e0h
            sjmp        dat_d2
            ;------------
data_kanan: mov         3ah,#00h
            mov         r0,#28h
```

```
            mov         r2,#08h
dat_k0      mov         r3,#08h
            mov         a,@r0
dat_k1      rlc         a
            jc          data_k
dat_k2      djnz        r3,dat_k1
            inc         r0
            djnz        r2,dat_k0
            ;semua dikalikan 4
            mov         a,3ah
            cjne        a,#01h,y1
            mov         a,#00h
            mov         3ah,a
            mov         f0h,a
y1          add         a,e0h
            da          a
            mov         3eh,a
            clr         a
            addc        a,#00h
            da          a
            mov         3fh,a
            ret
            ;data temporer
data_k      push        e0h
            mov         a,3ah
            add         a,#01h
            da          a
            mov         3ah,a
            pop         e0h
            sjmp        dat_k2
            ;
            end
;=============================================================
```

Tabel 3.4.8.List sub program conv_hex_BCD.

```
;=============================================================
;              SUB PROGRAM ConversiHex_BCD
;                 REVISI : Maret 2010
; Fungsi untuk mengkonversikan kode hexa ke BCD
; Input    : 40h(lsd)-43h(msd)=alamat buffer data biner 2 byte
; Output   : 50h (msb) s.d 51h (lsb) = alamat buffer data BCD
; Register : r0 = alamat sumber
;            r1 = alamat tujuan
;            r2 = counter byte BCD = r3+1 = 2+1 = 3
;            r3 = counter byte biner = n byte = 2
; Waktu Total : ……. M= …….. mS
;=============================================================
BINBCD:     push        e0h
            push        d0h
            push        03h
            push        02h
            push        01h
            push        00h
            ;--------------
            mov         r2,#03h
            mov         r3,#02h
            ;clear data buffer output BCD
            push        02h
            push        01h
            clr         a
clr_buf_bcd mov         @r1,a
            inc         r1
            djnz        r2,clr_buf_bcd
            pop         01h
            pop         02h
            ;clr buffer selesai
            ;proses konversi
pros_conv   push        03h
            mov         r3,#08 ;counter bit --> 1 byte = 8 bit
            mov         a,@r0
pros_con_1  clr         c
            rlc         a
            push        e0h
            lcall       adjus_conv
            pop         e0h
            djnz        r3,pros_con_1
            ;proses 1 byte selesai
            inc         r0
            pop         03h
            djnz        r3,pros_conv
            ;proses n byte selesai --> r2=n byte
            ;KELUAR
            pop         00h
            pop         01h
            pop         02h
            pop         03h
            pop         d0h
            pop         e0h
            ret
```

```
;-------------------------------------------------------------
adjus_conv: push        02h
            push        01h
adj_bcd     mov         a,@r1
            addc        a,@r1
            da          a
            mov         @r1,a
            inc         r1
            djnz        r2,adj_bcd
            pop         01h
            pop         02h
            ret

            end
;=============================================================
```

Tabel 3.4.9. List sub program Conv_hex_ASCII.

```
;=============================================================
;               SUB PROGRAM Conv_Hex_ASCII_Hex
: Fungsi    : Untuk mengkonversi kode hexa ke kode ASCII
; Input     : Sumber kode Hexa di register r4
; Output    : Tujuan kode ASCII di register r5
; Prinsip   : Memakai Tabel array di register dptr
; Register  :
;                r0   = pointer sumber
;                r1   = pointer tujuan
;                r2   = counter bit
;                dptr = pointer Tabel kode ASCII
;                a    = data sementara
; Table index: 0ff0h
; Total waktu = 191 M = 191 µS
;=============================================================
Conv_Hexa_ASCII:
            mov         r0,04h          ; #27h  r4=sumber
            mov         r1,05h       ;#3fh      r5=tujuan
            mov         r2,#08h
            mov         dptr,#0ff0h
            ;
hex_ascii   mov         a,@r0
            push        e0h
            orl         a,#f0h
            mov         dpl,a
            clr         a
            movc        a,@a+dptr
            mov         @r1,a
            dec         r1
            ; 1/2 byte lsd selesai
            pop         e0h
            swap        a
            orl         a,#f0h
            mov         dpl,a
            clr         a
            movc        a,@a+dptr
            mov         @r1,a
            ; 1 byte selesai
            dec         r1
            dec         r0
            djnz        r2,hex_ascii
            ret
;=============================================================
; Megkonversikan dari kode hexa hasil Scan_Code ke kode ASCCI
; Input     : alamat 20h ... 27h
; Output    : alamat 30h ... 3fh
; Register  : r0 = input
;             r1 = output
;             r2 = counter byte
; Table index: 0ff0h
; Total waktu = 191 M
;=============================================================
Conv_ASCII_Hexa:
            mov         r0,2dh      ;#30h
```

```
            mov         r1,#20h
            mov         r2,2ch      ;#08h
            mov         dptr,#0fe0h
            ;
ascii_hex   mov         a,@r0
            add         a,#b0h
            mov         dpl,a
            clr         a
            movc        a,@a+dptr
            swap        a
            mov         @r1,a
            inc         r0
            ; 1 byte msd selesai
            mov         a,@r0
            add         a,#b0h
            clr         a
            movc        a,@a+dptr
            orl         a,@r1
            mov         @r1,a
            ; 2 byte selesai
            inc         r1
            inc         r0
            djnz        r2,hex_ascii
      ;     pop         02h
      ;     pop         01h
      ;     pop         00h
            ret
;-------------------------------------------------------------
; alamat index Tabel
            org         0fd0h
tbl_ascii_hex:
            db
00h,01h,02h,03h,04h,05h,06h,07h,08h,09h
            org         0fe1h
            db          0Ah,0Bh,0Ch,0DH,0Eh,0Fh
            ;
            org         0ff0h
tbl_hex_ascii:
            db
30h,31h,32h,33h,34h,35h,36h,37h,38h,39h
            db          41h,42h,43h,44h,45h,46h
           end
;=============================================================
```

Tabel 3.4.10. List sub program meja uji

```
;=============================================================
;               SUB PROGRAM ME_Uji
; Fungsi    : Untuk mengendalikan motor di meja uji
; Input     : Kepad matrix
; Output    : P3.3 dan P3.4
; Call Subproram :
;           1. Delay 10mS
;           2. Delay 30mS
;           3. Delay 100mS
;=============================================================
t_motor     clr         p2.0        ;p2.2
            jnb         p2.4,m1
            setb        p3.3        ;p3.0
            jnb         p2.6,m2
            setb        p3.4        ;p3.1
            setb        p2.0        ;p2.2
            sjmp        out_motor
m1          jnb         p2.5,m1_cepat
            clr         p3.3        ;p3.0
            lcall       dly_10ms
            setb        p3.3        ;p3.0
            lcall       dly_100ms
            sjmp        t_motor
m1_cepat    clr         p3.3        ;p3.0
            lcall       dly_10ms
            lcall       dly_10ms
            lcall       dly_10ms
            lcall       dly_10ms
            setb        p3.3        ;p3.0
            lcall       dly_100ms
            sjmp        t_motor
m2          jnb         p2.5,m2_cepat
            clr         p3.4        ;p3.1
            lcall       dly_10ms
            setb        p3.4        ;p3.1
            lcall       dly_100ms
            sjmp        t_motor
m2_cepat    clr         p3.4        ;p3.1
            lcall       dly_10ms
            lcall       dly_10ms
            lcall       dly_10ms
            lcall       dly_10ms
            setb        p3.4        ;p3.1
            lcall       dly_100ms
            sjmp        t_motor
out_motor   ret

            end
;=============================================================
```

Lampiran 3 : Data Patent.

A. Tabel

Tabel 2.1. Jenis bahan dan metoda *tilt sensor* yang telah dipatenkan di *United State Patent* dari tahun 1965 s.d 2009. [9]

No.	Tahun	Bahan	Metoda	Kode Paten
1	Mar. 2,1965	*Electrolytic*	*Electrode*	3,171,213
2	June 20,1972	*Photo transistor*	*Element*	3,671,933
3	May 28,1974	*Mechanic*	*Element*	3,813,556
4	Apr. 22,1975	*Mercury*	*Electrode*	3,879,703
5	Oct. 14,1975	*Electrolytic*	*Electrode, mechanic*	3,911,592
6	Aug. 29,1978	*Liquid*	*Electrode, electronic*	4,110,609
7	May 22,1984	*Liquid*	*Ball, mechanic*	4,450,353
8	Oct. 22,1985	*Liquid*	*Electrode*	4,457,972
9	Jan. 24,1989	*Liquid*	*Electrode, acustic*	4,800,542
10	Feb. 7,1989	*Magnetic, fluid*	*Spiral magnetic*	4.803.426
11	Dec. 5,1989	*Magnetic*	*Magnetic permanent*	4.885.535
12	Aug. 4,1992	*Actuated*	*Ball*	5.136.127
13	Sep. 22,1992	*Mechanical*	*Micromechanical*	5.148.604
14	Jan. 18,1994	*Fluid*	*Electrode*	5.279.040
15	Oct. 4,1994	*Liquid*	*Electrode*	5.351.539
16	Dec. 13,1994	*Optical*	*Ball*	5.373.153
17	Oct. 10,1995	*Inductive*	*Magnetic*	5.546.013
18	Mar. 18, 1997	*Electrolytic*	*Electrode*	5.612.679
19	May 6, 1997	*Liquid*	*Electrode*	5.625.955
20	May 20,1997	*Electrolytic*	*Electrode*	5.630.280
21	Sep. 23, 1997	*Liquid*	*Electrode*	5.669.147
22	Oct. 28, 1997	*Temp. compensate*	*Electrode*	5.680.708
23	Nov. 4, 1997	*Strain gauge*	*Electrode*	5.684.456
24	May 12,1998	*Liquid*	*Electrode*	5,751,074
25	Jul. 7,1998	*Liquid*	*Ball*	5,777,290
26	Dec. 29,1998	*Electrolytic*	*Electrode*	5,852,,878
27	Jan. 4, 2000	*Photoelectric*	*Ball*	6,011,254
28	May 9, 2000	*Mechanic*	*Spring*	6,059,250
29	Jan. 9,2001	*Photoelectric*	*Ball*	US 6,172,357 B1
30	Jun. 26,2001	*Electrolytic*	*Electrode*	US 6,249,984 B1
31	Oct. 16,2001	*Magnetic*	*Ring Magnetic*	US 6,301,795 B1
32	Nov. 29,2001	*Capacitance*	*Electrode*	US 2001/0045019 A!

Lanjutan Tabel 2.1.

No.	Tahun	*.Bahan*	*Metoda*	Kode Paten
33	May 21, 2002	*Photoelectric*	*Ball*	US 6,392,233 B1
34	Sep. 3, 2002	*Capacitance*	*Electrode*	US 6,442,855 B2
35	Sep. 24, 2002	*Thermocouple*	*Electrode*	US 6,453,571 B1
36	Oct. 29, 2002	*Magnetic*	*Magnetic permanent*	US 6,470,580 B1
37	Oct. 29, 2002	*Magnetic*	*Ball*	US 8,472,864 B1
38	Nov. 14, 2002	*Strain gauge*	*Ball*	US 2002/0166756 A1
39	Jan. 2, 2003	*Capacitance*	*Electrode*	US 2003/0000095 A1
40	Apr. 8, 2003	*Magnetic*	*Ring Ball*	US 6,543,147 B2
41	Sep. 30, 2003	*Electrolytic*	*Ring Electrode*	US 6,625,896 B1
42	Jun. 19, 2003	*Electrolytic*	*Electrode*	US 2003/0110652 A1
43	Nov. 13, 2003	*Photoelectric*	*Electrode*	US 2003/0209654 A1
44	Nov. 20,2003	*Electrical*	*Electrode*	US 2003/0213136 A1
45	Feb. 10, 2004	*Electrolytic*	*Electrode*	US 6,688,013, B2
46	May 6, 2004	*Electrical*	*Ring Ball*	US 2004/0084290 A1
47	Oct. 12, 2004	*Electrolytic*	*Electrode*	US 6,802,132 B1
48	Oct. 28, 2004	*Photoelectric*	*Diffraction Grating*	US 2004/0213109 A1
49	Dec. 14, 2004	*Strength gauge*	*Field Strength*	US 6,831,456 B2
50	Dec. 16, 2004	*Ultrasonic*	*Micro sized*	US 2004/0251406 A1
51	Apr. 19, 2005	*Liquid*	*Electrode*	US 6,880,257 B1
52	Jul. 7, 2005	*Electrostatic Cap*	*Ring Electrode*	US 2005/0144794 A1
33	Aug. 23, 2005	*Strain gauge*	*MEMS Base*	US 6,934,662, B2
54	Sep. 15, 2005	*Strain gauge*	*Electrode*	US 2005/0198846 A1
55	Jan. 12, 2006	*Liquid*	*Electrode*	US 2006/0005407 A1
56	Jan. 24, 2006	*Electrostatic Cap.*	*Ring Electrode*	US 6,988,312 B2
57	Sep. 5, 2006	*Liquid*	*Electrode*	US 7,100,294 B1
58	Oct. 19, 2006	*Strain gauge*	*Electrode*	US 2006/0232736 A1
59	Aug. 14, 2007	*Strain gauge*	*Electrode*	US 7,254,897 B2
60	Oct. 11, 2007	*Optoelectronic*	*Electrode*	US 2007/0236685 A1
61	Nov. 27, 2007	*Optoelectronic*	*Electrode*	US 7,299,557 B2
62	Feb. 19, 2008	*Fluid*	*Electrode*	US 7,331,224 B2
63	Feb. 24, 2009	*Strain gauge*	*Electrode*	US 7,493,702 B2
64	Aug. 6, 2009	*Strain gauge*	*Electrode*	US 2009/0195243 A1
65	Aug. 18, 2009	*Photoelectric*	*Diffraction Grating*	US 7,577,076 B2

B. Data US Patent.

B1. Nov. 13, 2003

US 20030209654A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0209654 A1**

Smith (43) **Pub. Date: Nov. 13, 2003**

(54) **ELECTRONIC TILT SWITCH AND INTEGRATED LIGHT MODULE**

(75) Inventor: **William Colie Smith**, McMinnville, TN (US)

Correspondence Address:
JAMES D. STEVENS
REISING, ETHINGTON, BARNES, KISSELLE, ET AL
P.O. BOX 4390
TROY, MI 48099 (US)

(73) Assignee: **FEDERAL-MOGUL WORLD WIDE, INC.**, Southfield, MI

(21) Appl. No.: **10/144,108**

(22) Filed: **May 13, 2002**

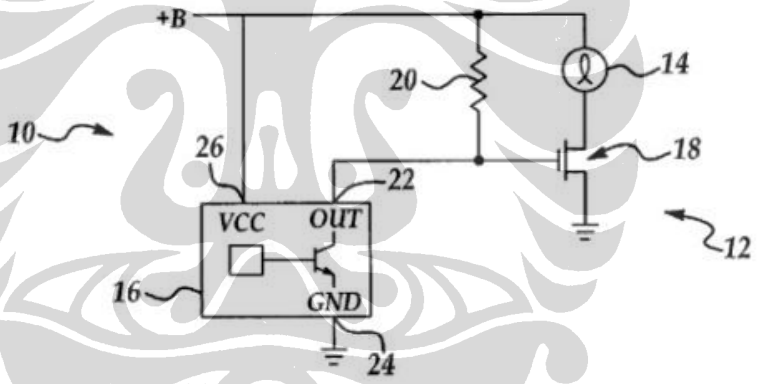
Publication Classification

(51) **Int. Cl.**[7] **H01J 40/14**

(52) **U.S. Cl.** ..... **250/214 SW**

(57) **ABSTRACT**

An electronic tilt switch for controlling an electrical circuit such as a vehicle hood lamp circuit according to the angle of inclination of the tilt switch. The tilt switch includes a housing, a sliding element, and a solid state electronic sensor, which can comprises a Hall effect sensor or a photo-optical interrupter. The sliding element moves within an axial bore of the housing according to the angle of inclination of the tilt switch. The solid state electronic sensor is mounted to one axial end of the housing and is capable of determining when the sliding element slides past a predetermined position within the bore. This allows the tilt switch to determine whether the vehicle hood has been opened or shut, and is used to turn the lamp on or off accordingly. The housing, sliding element, sensor, and associated electronics can all be environmentally sealed within a single package, which can additionally include one or more molded in connectors. Also disclosed is an integrated tilt switch/lighting module in which one or more light sources are integrated into the sealed package.



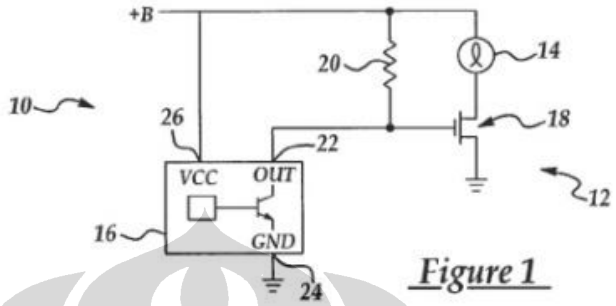


Figure 1

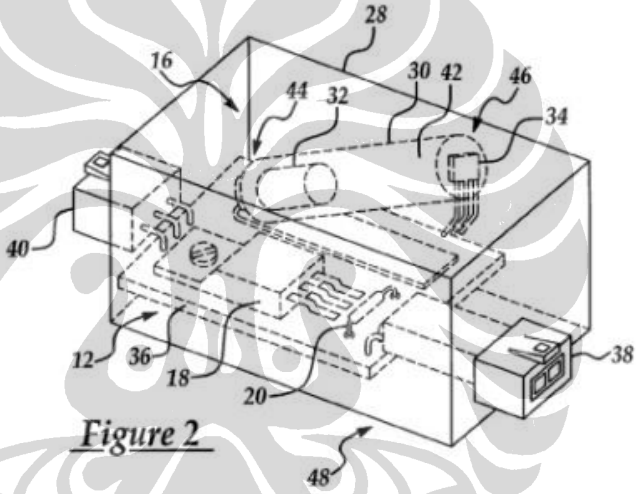


Figure 2

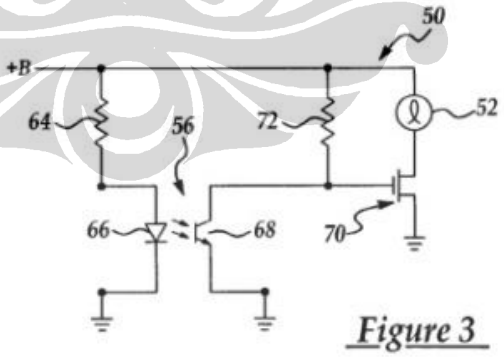


Figure 3

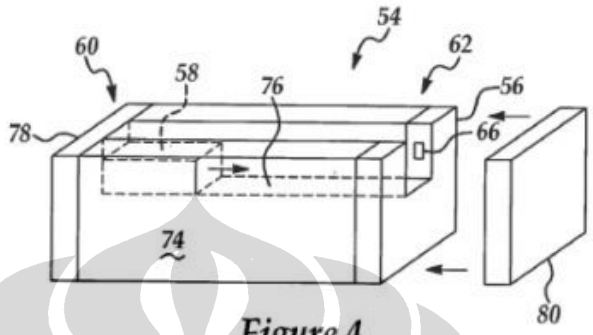


Figure 4

Figure 5

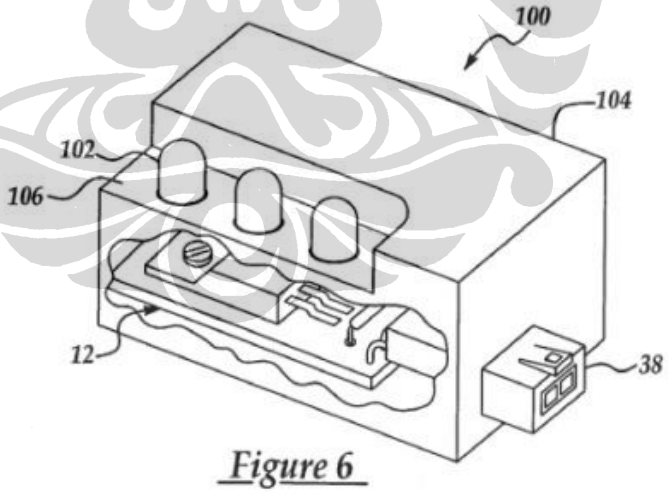


Figure 6

ELECTRONIC TILT SWITCH AND INTEGRATED LIGHT MODULE

TECHNICAL FIELD

[0001] The present invention relates generally to tilt switches used in electronic circuits and to such switches used in automotive lighting applications; for example, as a trunk or hood switch used to activate a light when a vehicle trunk or hood is opened.

BACKGROUND OF THE INVENTION

[0002] Traditionally, tilt switches made from mercury capsules have been used in a wide variety of applications. For instance, mercury switches have been utilized in lighting circuits that control lights located on the underside of vehicle hoods. When the hood is opened, and hence tilted beyond a certain angle, the switch would turn on a light. A non-automotive example is in simple residential thermostats where the mercury switch is mounted on a spiral flat spring and used to switch on and off a HVAC system as the spring thermally expands or contracts. However, in an effort to limit mercury contamination due to improper disposal of the mercury switches at the end of their useful life, it has become desirable to find a tilt switch design which does not utilize mercury. This has resulted in the development of different tilt switch designs that use non-mercury based components, such as permanent magnet/reed switch combinations, to sense changes in inclination.

[0003] For example, U.S. Pat. No. 4,820,888 issued Apr. 11, 1989 to Shields discloses the use of a magnetically operated tilt switch assembly that generally includes a reed switch, an elongated cavity, and a sliding magnet. When the switch assembly tilts beyond a predetermined angle, the magnet slides within the cavity towards and away from the reed switch, which is located at one axial end of the cavity. The proximity of the magnet to the reed switch controls the operation of the switch assembly.

[0004] Similarly, U.S. Pat. No. 3,564,171 issued Feb. 16, 1971 to Hammond discloses a magnetic filed activated tilt switch assembly having an elongated cavity for receiving a sliding magnet. The magnet comprises a bar magnet sandwiched between two ferrous balls which are wide enough that the magnet does not come in direct contact with the sides of the cavity. As the switch housing is tilted beyond a predetermined angle, the magnet slides towards or away from a reed switch mounted on the topside of the cavity, thereby closing the switch when the magnet is in close proximity and opening the switch when the magnet is at the other end of the cavity.

[0005] U.S. Pat. No. 3,601,729 issued Aug. 24, 1971 to Hierta also teaches the use of a tilt switch assembly incorporating a reed switch to detect inclination of the device beyond a certain angle. The tilt switch assembly that includes a switch housing having an elongated V-shaped cavity which contains a cylindrical magnetic member. The apex of the cavity is closest in distance to a reed style switch. Accordingly, when the switch housing is in an upright (non-tilted) orientation, the magnetic flux from the magnetic member is strong enough to pull the two contacts of the reed switch together, thereby closing the switch. As the switch housing is tilted, the magnetic member rolls away from the apex, thereby becoming sufficiently spaced from the reed switch and unable to pull the switch contacts together.

[0006] U.S. Pat. No. 5,209,343 issued May 11, 1993 to Romano et al. discloses a magnetic field sensing tilt switch, shown in **FIG. 21**, that generally comprises a cup shaped housing, a magnetized ball, and a magnetic sensor switch. As the switch assembly is tilted, the ball rolls towards and away from the magnetic field sensing switch, thus causing the switch to change states according to the position of the ball within the housing.

[0007] Further examples of tilt switch assemblies utilizing reed switches can be found in U.S. Pat. Nos. 5,256,839, 5,477,428, 5,669,696, and 5,798,912. Other magnetically-actuated switch designs have been proposed using Hall effect sensors rather than reed switches. See, for example, U.S. Pat. No. 5,373,125, issued Dec. 13, 1994 to Ford et al. This patent discloses a magnetic field sensing tilt switch assembly wherein a pivotal member having a permanent magnet pivots within a switch housing containing a Hall effect sensor. When the switch assembly is in an upright position the magnetic field is aligned with the sensor, however, when the switch assembly is tilted, the magnet and the sensor become unaligned, which results in the switch changing states.

[0008] Apart from magnetic switches, other non-mercury based technologies such as photoelectric detectors have been used in tilt switches. For example, U.S. Pat. No. 5,202,559 utilizes a photo-optical interrupter and a box-like chamber and ball to detect a tilt beyond a predetermined angle. Two of the four interior side walls of the box have light emitting means mounted such that they communicate with light detecting means mounted on the opposing interior wall, and each interior side wall has a semi-spherical recess designed to receive the ball. When the switch is tilted beyond a predetermined angle, the ball rolls downwards and seats on the lowest recess, thereby obstructing the light emitting and detecting means associated with that recess and changing the state of the switch.

[0009] Likewise, U.S. Pat. No. 5,209,343 discloses a photooptical tilt switch, seen in **FIG. 18**, that includes a cylindrical housing containing a weighted ball. At one axial end of the housing is a light transmitter and light receiver mounted in opposing fashion along the interior walls of the housing. Thus, when the switch assembly is tilted, the ball rolls to the end of the cylindrical housing having the light sensing components and prevents the optical receiver from receiving the emitted light, thus causing the switch to change states. Additional tilt switch designs incorporating photo-optical interrupters can be seen in U.S. Pat. Nos. 5,373,153, 6,011,254, and 6,140,635.

SUMMARY OF THE INVENTION

[0010] In accordance with one aspect of the present invention, there is provided an electronic tilt switch capable of controlling the state of an electrical circuit according to the angle of inclination of the switch. The switch includes a housing having an elongated bore that extends internally within the housing, a sliding element confined within the bore, and a solid state electronic sensor located adjacent the bore. The sliding element can slide along the length of the bore into and out of a predetermined position within the bore. The sensor is capable of sensing when the angle of

inclination of the switch exceeds a predetermined amount by determining when the sliding element is located at the predetermined position.

[0011] The solid state electronic sensor can be any of a number of different suitable devices, such as, for example, a Hall effect sensor or photo-optical interrupter. In the case of a Hall effect sensor, the sliding element can be a permanent magnet that produces a magnetic field which is sensed by the Hall effect sensor when the magnet slides into the predetermined position within the bore. Where a photo-optical interrupter is used, the sensor is located such that it establishes a light path that extends across at least a portion of the elongated bore at the predetermined position. The sliding element can then be made from an opaque material such that it obstructs the light path when it moves into the predetermined position. The tilt switch can include an electrical connector for easy connection to the switch, as well as an electrical circuit interconnecting the sensor and connector. Preferably, the housing, sliding element, sensor, and electrical circuit are located internally within an environmentally sealed package.

[0012] In accordance with another aspect of the present invention, there is provided an integrated electronic tilt switch and lighting module that can be used as, for example, an under-the-hood lamp for a vehicle to illuminate the engine compartment when the vehicle hood is raised. The module comprises a housing having an elongated bore that extends internally within the housing, a sliding element confined within the bore, a solid state electronic sensor, a light source, and an electrical circuit located connected to both the sensor and light source. The sensor detects when the sliding element moves to a predetermined position within the bore, and the circuit responds to the sensor to switch on (or off) the light source. The housing, sliding element, sensor, and electrical circuit are all located internally within an environmentally sealed package. Preferably, the housing includes an integral connector located at an exterior surface of the package.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Preferred exemplary embodiments of the invention will hereinafter be described in conjunction with the appended drawings, wherein like designations denote like elements, and wherein:

[0014] **FIG. 1** is a schematic view of an electronic tilt switch constructed in accordance with a first embodiment of the present invention and utilizing a Hall effect sensor in the tilt switch to control energization of a lamp;

[0015] **FIG. 2** is a perspective view of the tilt switch of **FIG. 1**;

[0016] **FIG. 3** is a schematic view of an electronic tilt switch constructed in accordance with a second embodiment of the present invention and utilizing a photo-optical interrupter in the tilt switch to control energization of a lamp;

[0017] **FIG. 4** is a perspective view of the tilt switch of **FIG. 3**;

[0018] **FIG. 5** is a perspective view of an opaque disk that can be used in the tilt switch of **FIGS. 3 and 4**; and

[0019] **FIG. 6** is a perspective view of an integrated electronic tilt switch and lighting module wherein portions of the internal components of the module can be seen in cutaway.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0020] Referring now to **FIG. 1**, there is seen a schematic view of a lighting module **10** that includes an electronic circuit **12** that controls the operation of a lamp **14** according to the angle of inclination of an electronic tilt switch **16**. The lighting module **10** can be used in any of a number of different applications, and the following discussion will be directed to its use for an automotive under-the-hood application in which the module **10** is located under a vehicle's hood (not shown) so that the lamp **14** is energized when the hood is raised, and is extinguished when the hood is lowered and closed. Generally, the tilt switch **16** is mounted to the underside of the vehicle hood such that pivotal upward movement of the hood causes the tilt switch to change its angle of inclination. When the hood is opened, the angle of inclination of the tilt switch **16** exceeds a certain angle sufficient for the switch to change states, and the circuit **12** is responsive to this change in state to energize the lamp **14**. When the hood is closed, the angle of inclination of the tilt switch **16** falls below the angle required for activation, and the tilt switch turns the hood lamp **14** to off via the circuit **12**. As shown, power is supplied from the vehicle battery via a supply rail +B and is supplied both to the circuit **12** and tilt switch **16**.

[0021] The electronic tilt switch **16** of **FIG. 1** is a Hall effect sensor type switch that generally includes an output **22**, a ground connection **24**, and a voltage source input **26**. As will be appreciated by those skilled in the art, the following description of the Hall effect sensor is directed to the logic involved in operating the lamp based on tilt switch position and is not meant to describe any particular commercially available Hall effect sensor. Rather, those skilled in the art will know how to obtain the desired switching of the lamp using different Hall effect sensor arrangements together with suitable circuitry to implement the necessary switching logic. For the particular arrangement shown, when the tilt switch **16** is oriented such that it is in its "off" position, its internal transistor switch is biased on, which connects its output **22** to ground **24**. Output **22** is connected to the gate of an N-channel MOSFET transistor **18**, such that pulling its gate to ground causes the transistor to function in an "off" state. Accordingly, hood lamp **14** is disconnected from voltage source +B and therefore does not illuminate. Once the hood is opened and pivots such that the tilt switch is oriented beyond the minimum number of degrees required, tilt switch **16** switches off its transistor providing a high impedance state on output **22**. This allows application of the supply voltage +B to the gate of transistor **18** via a pullup resistor **20**, thereby causing transistor **16** to operate in an "on" state. In this state, the transistor is conductive and connects the hood lamp **14** to the voltage source +B, thus turning on the lamp and illuminating the surrounding area. Again, the illustrated circuitry is but one approach to utilizing the Hall effect sensor switch and it will be understood that any of a number of other circuits can be designed to utilize a Hall effect sensor type tilt switch without departing from the scope of the present invention.

[0022] Referring now to **FIG. 2**, the Hall effect sensor tilt switch **16** is seen in further detail and generally includes an elongated, cylindrical housing **30**, a magnetic sliding element **32**, a Hall effect sensor **34**, and the electrical circuit **12** which is mounted on a printed circuit board **36**. All of these

components are mounted within an environmentally sealed package **28** to protect them against moisture and contaminants. The tilt switch **16** also includes an input connector **38** and an output connector **40** mounted at opposed exterior surfaces of the package **28**. Housing **30** can be a rigid cylindrical plastic component having an elongated cylindrical bore **42** extending from a first axial end **44** to a second axial end **46**. Sliding element **32** is shaped to be received by the cylindrical bore **42** such that it is freely capable of sliding along the longitudinal axis of the bore. For this purpose, the sliding element **32** and/or the bore surface itself can be provided with a lubricant, such as Teflon™ or graphite, that is either provided as a part of the components themselves (e.g., a coating) or separately applied during assembly of the device.

[0023] Housing **30** is mounted to the printed circuit board (PCB) **36** and inclines upwardly away from the PCB such that the sliding element **32** moves to the first axial end **44** under the force of gravity when the bottom surface **48** and, hence, the PCB **36**, is placed in a substantially horizontal orientation. As will be appreciated, when the package **28** is tilted through an angle sufficient to raise the first axial end **44** well above the second axial end **46**, the sliding element **32** will move to the second axial end **46** under the force of gravity. To detect this movement, the Hall effect sensor **34** is located adjacent the housing **30** at a predetermined position along the length of the bore which in the illustrated embodiment is at the second axial end **46**. The sliding element **32** is a permanent magnet that produces a magnetic field which impinges upon the Hall effect sensor **34** when the sliding element moves to the predetermined position at the second axial end **46**. The Hall effect sensor **38** utilizes the magnetic field produced by the sliding magnetic element to induce a voltage according to a phenomenon known as the Hall effect, which is commonly known in the art. When the magnetic field exceeds a predetermined amount, the switch is in an "on" state and turns on lamp **12** accordingly. When the magnetic field produced by sliding magnetic component **32** is less than that required for activation of the tilt switch, then the switch operates in an "off" state and the lamp **14** is not supplied with power. The Hall effect sensor **34** can be electrically connected to the circuit **12** by means of jumper wires (not shown); and these wires correspond to the connections **22**, **24**, and **26** previously discussed.

[0024] In operation, when tilt switch **16** is in a generally horizontal orientation, as when the vehicle hood is closed thus making housing **30** inclined downwardly towards its first end **44**, the sliding element **32** is generally located at the axial end **44** opposite the Hall effect sensor **34**, as shown. Because the sliding element is separated from the Hall effect sensor by the axial length of the bore, the magnetic field experienced by the sensor is too weak to activate the tilt switch. Once the switch **16** is tilted enough to cause the sliding element **32** to slide to the second axial end **46** of the housing under the force of gravity, the field produced by the sliding element and sensed by the Hall effect sensor is strong enough to cause the switch to change states, thereby enabling circuit **12** to energize the lamp **14**. The lamp remains on until the angular inclination of the tilt switch **16** again approaches horizontal, at which time magnetic member **32** slides away from the Hall effect sensor **34** to the opposite end **44** of the housing, thereby turning the switch and the lamp off.

[0025] It should be noted that tilt switch **14** could operate in a converse manner. That is, the electronic tilt switch could be designed such that it operates in an "off" state, thus turning off lamp **14**, when sliding element **32** is located at the second axial end **46**. In this embodiment, pivoting the tilt switch beyond a certain number of degrees would cause the sliding element **32** to slide away from the Hall effect sensor **34** and towards the first axial end **44**, thereby turning both switch **16** and the lamp **14** to an "on" state.

[0026] Preferably, connectors **38** and **40** are integrally and rigidly connected to sealed package **28**, and these connectors can include latch mechanisms to permit secure mating and retention of a connection plug from a wiring harness or the like. The connectors can also be keyed to prevent improper insertion of the mating plugs. The components that are environmentally sealed within package **28** can be assembled to the printed circuit board **36** which can then be inserted into an open side of package **28** and, once positioned within the package, can be potted in place to seal them within the package. Other techniques for environmentally sealing these components within package **28** will be known to those skilled in the art.

[0027] Referring now to **FIGS. 3 and 4**, there is seen a schematic view of an electronic circuit **50** incorporating an alternative embodiment **54** of the tilt switch of the present invention, again used for a vehicle hood lighting circuit to illuminate a lamp **52**. This embodiment is the same as that of **FIGS. 1 and 2**, with the exceptions noted below. In particular, the Hall effect sensor is replaced by a photo-optical interrupter **56** and the tilt switch mechanical design is altered accordingly to utilize this photo interrupter to detect when the sliding element **58** is situated at the predetermined position (second axial end **62**). Schematically, as shown in **FIG. 3**, the circuit design is modified to include the photo interrupter **56** in place of the Hall effect sensor with a current limiting resistor **64** in series with the photo interrupter's power input. The photo interrupter **56** is an infrared semiconductor switch having a photo diode **66** and a photo transistor **68**. Light from the photo diode **66** traverses a gap between the two components and impinges upon the photo transistor **68**, thereby allowing current flow through the transistor between its collector and emitter. In this state, the gate of the MOSFET **70** is clamped to ground, holding it and lamp **52** in an "off" state. If the light from photodiode **66** is obstructed from reaching the photo transistor **68**, then the transistor **68** switches off, allowing the MOSFET **70** to be biased on by way of the pullup resistor **72**. As a result, the lamp **52** is energized.

[0028] As shown in **FIG. 4**, the electronic tilt switch **54** incorporating the photo-optical interrupter **56** is seen in further detail. This tilt switch **54** can be substituted into the overall module shown in **FIG. 2** in place of the Hall effect sensor and its associated housing and sliding element. For tilt switch **54**, the sliding element **58** need not be a magnet, but instead is optically opaque to the light transmitted by the photo diode **66**. As shown, housing **74** includes a axial bore **76** extending along its longitudinal length to accommodate the sliding element **58**. In the particular embodiment seen, the axial bore is open upon its upper side; however, the bore could instead be a closed bore such that it encloses the sliding element within the housing. At the first axial end **60** of the housing **74**, there is provided an end cap **78** to prevent the sliding element **58** from sliding out of the bore **76**. Of

course, a similar top cap (not shown) could be used to prevent the sliding piece from coming up out of the channel. The photo interrupter **56** is located at the second axial end **62** of the bore, and includes a physical gap between the photo diode **66** and photo transistor **68** (not shown) which permits the sliding element **58** to move in between the photo diode and photo transistor to obstruct the light path that otherwise normally exists between these components. A second end cap **80** can be used at this second end **62** to keep the sliding element **58** in the bore.

[0029] In a manner of operation similar to the Hall effect sensor tilt switch previously described, the electronic photo-optical interrupter tilt switch **54** is in an "off" state when sliding member **58** does not interfere with the light path established at the first axial end of the housing and is in an "on" state when the switch is tilted such that the sliding member slides into a light path obstructing position. The non-interfering, "off" state condition is shown in **FIG. 4**. In this position, a light path is established between photo diode **66** and photo transistor **68** (not shown) such that lamp **52** in the corresponding circuit is not powered and therefore is off. Once the vehicle hood is pivoted beyond a certain angle of inclination, sliding element **58** moves from the first axial end **60** of the housing to the second axial end **62**. When the sliding element reaches the second end, it obstructs the light path such that the photo transistor **68** is unable to receive light emitted from the photo diode **66**. As previously described in conjunction with **FIG. 3**, the obstruction of the light path turns on lamp **52** such that it illuminates the surrounding area. When the vehicle hood is closed, the electronic switch assumes a position that causes the sliding element **58** to slide to the first end **60**, thereby turning lamp **52** off again. Accordingly, the electronic switch **54** is capable of controlling an associated circuit **50**, such as a vehicle engine compartment lighting circuit, by its angle of inclination.

[0030] Again, it should be noted that tilt switch **54** could operate in an opposite manner. That is, the electronic tilt switch could be designed such that it operates in an "off" state, thus turning off lamp **52**, when the sliding element **58** is located at second axial end **62**. In this embodiment, pivoting the tilt switch beyond a certain number of degrees would cause the sliding element to slide away from the photo-interrupter towards the first end **60**, thereby turning both switch **54** and lamp **52** to an "on" state.

[0031] In lieu of the sliding element **58**, an opaque disk **82** can be used, as shown in **FIG. 5**. This disk is sized in diameter and thickness to fit within the bore such that it can roll between the first and second ends **60**, **62** depending upon the angle of inclination of the tile switch.

[0032] Referring to yet another embodiment of the present invention, **FIG. 6** illustrates an electronic tilt switch/lighting module **100** which incorporates electronic circuit **12** and Hall effect sensor tilt switch **16** of **FIGS. 1 and 2** along with an integral light source such as three white LEDs **102**. Incandescent or other types of light sources can be used instead of the LEDs. Also, the photo interrupter tilt switch **54** or other suitable tilt switch can be utilized in lieu of switch **16**. Integrating the lamp together with the tilt switch assembly into a single module makes attaching the single module to the vehicle easier than attaching these components separately along with their associated wiring. This approach also eliminates the need for the output connector **40**. The tilt switch (along with its housing, sliding element, and sensor) and the electronic circuit, including its printed circuit board, are all environmentally sealed within a single package **104**. Furthermore, both the LEDs and connector **38** can be sealed in place on the housing, making all of the internal wiring connections completely sealed against moisture and contaminants.

[0033] As shown, the LEDs **102** can be located in a recessed area **106** located on the top surface of package **104**. They are wired to the printed circuit board prior to the entire contents of the package **104** being potted or otherwise sealed inside the package. If desired, the surface of the recessed area **106** can be utilized as a reflector with a suitable geometry and/or coating to control the distribution of light emitted from the module **100**.

[0034] It will thus be apparent that there has been provided in accordance with the present invention an electronic tilt switch and an integrated electronic tilt switch and lighting module which achieves the aims and advantages specified herein. It will of course be understood that the foregoing description is of preferred exemplary embodiments of the invention and that the invention is not limited to the specific embodiments shown. Various changes and modifications will become apparent to those skilled in the art and all such variations and modifications are intended to come within the spirit and scope of the appended claims.

I claim:

1. An electronic tilt switch capable of controlling the state of an electrical circuit according to the angle of inclination of said switch, comprising:

a housing having an elongated bore that extends internally within said housing;

a sliding element confined within said bore and being capable of sliding into and out of a predetermined position within said bore; and

a solid state electronic sensor located adjacent said bore, wherein said sensor is capable of sensing when the angle of inclination of said switch exceeds a predetermined amount by determining when said sliding element is located at said predetermined position within said bore.

2. An electronic tilt switch as defined in claim 1, wherein said solid state electronic sensor is a Hall effect sensor capable of detecting a magnetic field and wherein said sliding element is a magnetic sliding element that produces a magnetic field.

3. An electronic tilt switch as defined in claim 1, wherein said solid state electronic sensor is a photo-optical interrupter capable of establishing a light path extending across at least a portion of said elongated bore at said predetermined position, and wherein said sliding element comprises an opaque material such that said sliding element is capable of obstructing said light path when said sliding element is located at said predetermined position.

4. An electronic tilt switch as defined in claim 1, further comprising an electrical connector and an electrical circuit interconnecting said sensor and said connector.

5. An electronic tilt switch as defined in claim 4, further comprising an environmentally sealed package, wherein said housing, sliding element, sensor, and electrical circuit

are environmentally sealed within said package and wherein said connector is located at an exterior surface of said package.

6. An electronic tilt switch as defined in claim 4, wherein said housing and electrical circuit are both mounted on a printed circuit board that is environmentally sealed within said package.

7. An electronic tilt switch capable of controlling the state of an electrical circuit according to the angle of inclination of said switch, comprising:

a housing having an elongated bore that extends internally within said housing;

a magnetic sliding element located within said bore and being capable of sliding along the length of said bore according to the angle of inclination of said switch; and

a solid state electronic Hall effect sensor located at one end of said bore, wherein said sliding element is located away from said one end of said bore when said switch is at a first angle of inclination and thereafter slides to said one end of said bore when said switch is at moved to a predetermined second angle of inclination, and wherein said Hall effect sensor is capable of sensing when said switch is at said second angle of inclination by sensing the magnetic field created by said magnetic sliding element when it moves to said one end of said bore.

8. An electronic tilt switch as defined in claim 7, wherein said tilt switch further includes a voltage source input, a ground connection, and an output.

9. An electronic tilt switch as defined in claim 8, wherein said output is coupled to said ground connection when said tilt switch is in an "off" state and said output is coupled to said voltage source input when said tilt switch is in an "on" state.

10. An electronic tilt switch as defined in claim 9, wherein said output is connected to a transistor which controls the state of the electrical circuit.

11. An electronic tilt switch as defined in claim 7, wherein said tilt switch is mounted to the underside of a vehicle hood.

12. An electronic tilt switch capable of controlling the state of an electrical circuit according to the angle of inclination of said switch, comprising:

a housing having an elongated bore that extends internally within said housing;

an opaque element located within said bore and being capable of moving along the length of said bore according to the angle of inclination of said switch; and

a solid state photo-optical interrupter located at one axial end of said housing and capable of establishing a light path that extends across at least a portion of said elongated bore, wherein said photo-optical interrupter is capable of sensing when the angle of inclination of said switch exceeds a predetermined amount by determining when said opaque element obstructs said light path.

13. An electronic tilt switch as defined in claim 12, wherein said opaque element is a sliding element that slides along the length of said bore depending upon the angle of inclination of said housing.

14. An electronic tilt switch as defined in claim 12, wherein said opaque element is a disk that rolls along the length of said bore depending upon the angle of inclination of said housing.

15. An electronic tilt switch as defined in claim 12, wherein said tilt switch is mounted to the underside of a vehicle hood.

16. An integrated electronic tilt switch and lighting module, comprising:

a housing having an elongated bore that extends internally within said housing;

a movable element confined within said bore and being capable of moving into and out of a predetermined position within said bore;

a solid state electronic sensor located adjacent said bore such that said sensor can detect when said movable element is located at said predetermined position;

a light source;

an electrical circuit connected to both said sensor and said light source, said circuit being responsive to said sensor to selectively energize or de-energize said light source based on the angle of inclination of said housing; and

an environmentally sealed package, wherein said housing, movable element, sensor, and electrical circuit are environmentally sealed internally within said package.

17. An integrated electronic tilt switch and lighting module as defined in claim 16, wherein said solid state electronic sensor is a Hall effect sensor capable of detecting a magnetic field.

18. An integrated electronic tilt switch and lighting module as defined in claim 17, wherein said movable element is a permanent magnet sliding element.

19. An integrated electronic tilt switch and lighting module as defined in claim 16, wherein said solid state electronic sensor is a photo-optical interrupter capable of establishing a light path extending across at least a portion of said elongated bore at said predetermined position.

20. An integrated electronic tilt switch and lighting module as defined in claim 19, wherein said movable element is an opaque sliding element.

21. An integrated electronic tilt switch and lighting module as defined in claim 19, wherein said movable element is an opaque disk that rolls along said bore.

22. An integrated electronic tilt switch and lighting module as defined in claim 16, wherein said light source comprises a vehicle compartment light.

23. An integrated electronic tilt switch and lighting module as defined in claim 16, further comprising an integral connector located on an exterior surface of said sealed package and being electrically connected to said circuit within said package.

* * * * *

B.2. Aug. 18, 2009

US007577076B2

(12) **United States Patent**
Ogata et al.

(10) **Patent No.: US 7,577,076 B2**
(45) **Date of Patent: Aug. 18, 2009**

(54) **TILT SENSOR USING DIFFRACTION GRATING**

(75) Inventors: **Tetsuya Ogata**, Tokyo (JP); **Shigeru Oohchida**, Tokyo (JP); **Junichi Kitabayashi**, Kanagawa (JP); **Suguru Douwaki**, Kanagawa (JP)

(73) Assignee: **Ricoh Company, Ltd.**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 676 days.

(21) Appl. No.: **10/798,362**

(22) Filed: **Mar. 12, 2004**

(65) **Prior Publication Data**

US 2004/0213109 A1 Oct. 28, 2004

(30) **Foreign Application Priority Data**

Mar. 14, 2003 (JP) 2003-070509
Aug. 12, 2003 (JP) 2003-292010
Sep. 30, 2003 (JP) 2003-340394

(51) **Int. Cl.**
G11B 7/135 (2006.01)

(52) **U.S. Cl.** **369/112.03**; 369/53.19

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,015,835 A 5/1991 Ohuchida et al.
5,065,380 A * 11/1991 Yokota 369/44.12
5,101,389 A 3/1992 Ohuchida et al.
5,107,359 A 4/1992 Ohuchida
5,144,584 A 9/1992 Inada et al.
5,231,520 A 7/1993 Ohuchida
5,243,583 A 9/1993 Ohuchida et al.
5,257,131 A * 10/1993 Yoshida et al. 359/485
5,278,817 A 1/1994 Maeda et al.
5,281,802 A 1/1994 Kitabayashi
5,325,348 A 6/1994 Maeda et al.
5,428,588 A 6/1995 Ohuchida
5,487,058 A 1/1996 Kitabayashi
5,684,779 A 11/1997 Ohuchida et al.
5,956,302 A 9/1999 Maeda et al.
6,184,984 B1 2/2001 Lee et al.
6,584,060 B1 6/2003 Oohchida

(Continued)

FOREIGN PATENT DOCUMENTS

JP 57179954 A 11/1982

(Continued)

OTHER PUBLICATIONS

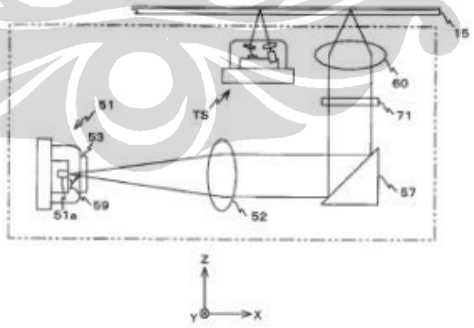
J. Koyama, et al. "Light Wave Electronics", *Corona*, Jul. 15, 1993, pp. 117-132.

Primary Examiner—Joseph H Feild
Assistant Examiner—Tawfik Gotiya
(74) *Attorney, Agent, or Firm*—Dickstein Shapiro LLP

(57) **ABSTRACT**

A tilt sensor for detecting information related to a tilt of an object to a reference plane is disclosed. The tilt sensor includes a diffraction element disposed that diffracts an incident light at diffraction efficiency depending on the incident angle, and a photo detector that detects a diffraction light diffracted by said diffraction element and outputs an photoelectric signal. Since the diffraction efficiency of the diffraction element changes as the incident angle changes, the intensity of the diffraction light from the diffraction element changes. As a result, the photoelectric signal output from the photo detector contains the information related to the tilt of the object.

5 Claims, 60 Drawing Sheets



U.S. PATENT DOCUMENTS

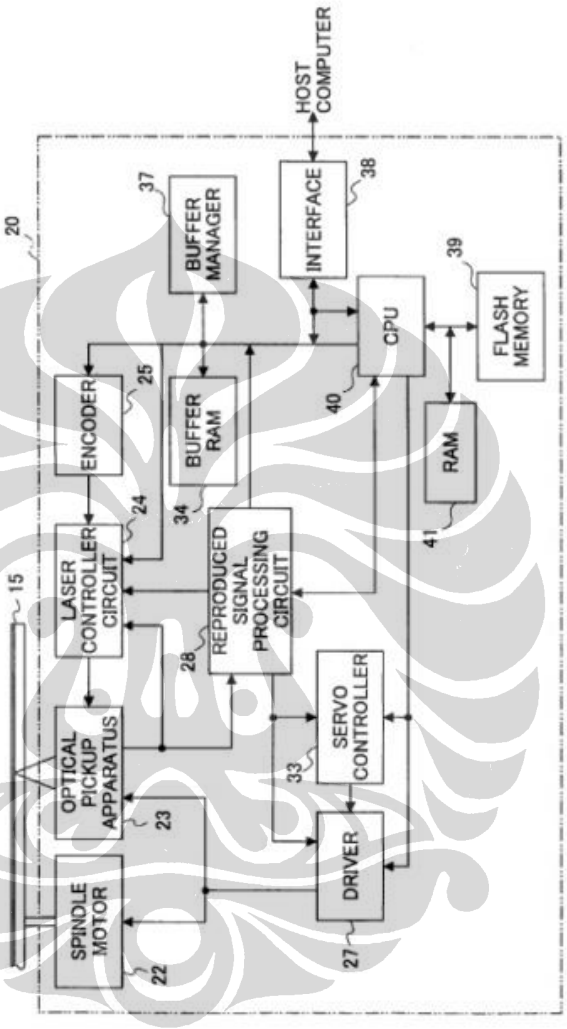
6,738,324	B2 *	5/2004	Saimi et al.	369/44.37
6,876,621	B2	4/2005	Ohuchida et al.	
2002/0018432	A1	2/2002	Ohuchida	
2002/0018433	A1	2/2002	Ohuchida	
2002/0041542	A1	4/2002	Sano et al.	
2002/0093902	A1	7/2002	Hirai et al.	
2003/0072047	A1	4/2003	Funato et al.	
2003/0193876	A1	10/2003	Oohchida et al.	
2003/0214898	A1	11/2003	Ogata et al.	
2004/0013076	A1	1/2004	Funato et al.	

FOREIGN PATENT DOCUMENTS

JP	61-86826	6/1986	
JP	62-22727	2/1987	
JP	03-037837	2/1991	
JP	02-68734	3/1991	
JP	05006561	A	1/1993
JP	05006562	A	1/1993
JP	07-270723	10/1995	
JP	08320217	A	12/1996
JP	2002-190125	7/2002	
JP	2002-543381	12/2002	
JP	2003-043254	2/2003	
JP	2003-195023	7/2003	
JP	2003-203379	7/2003	
JP	2003-257051	9/2003	

* cited by examiner

FIG. 1



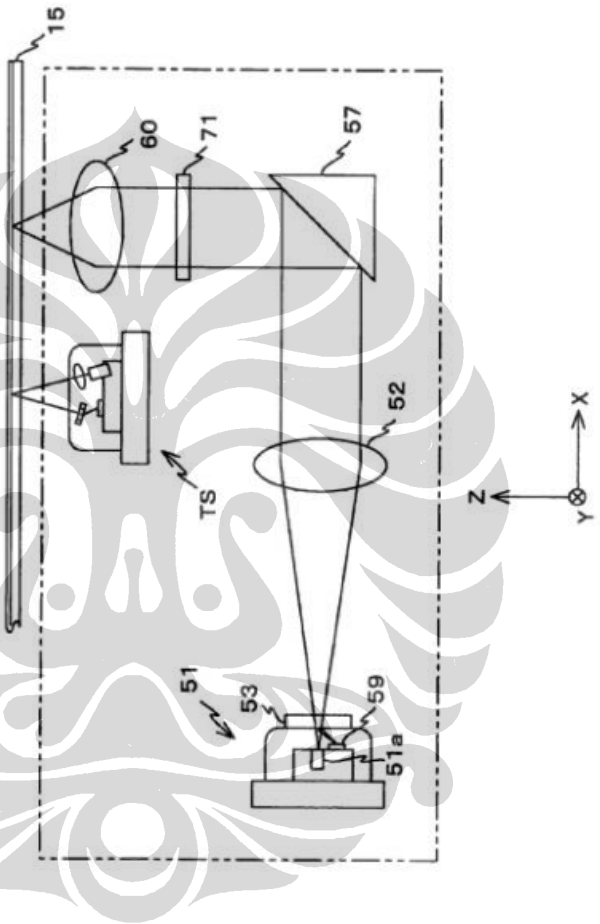


FIG.2

FIG.3

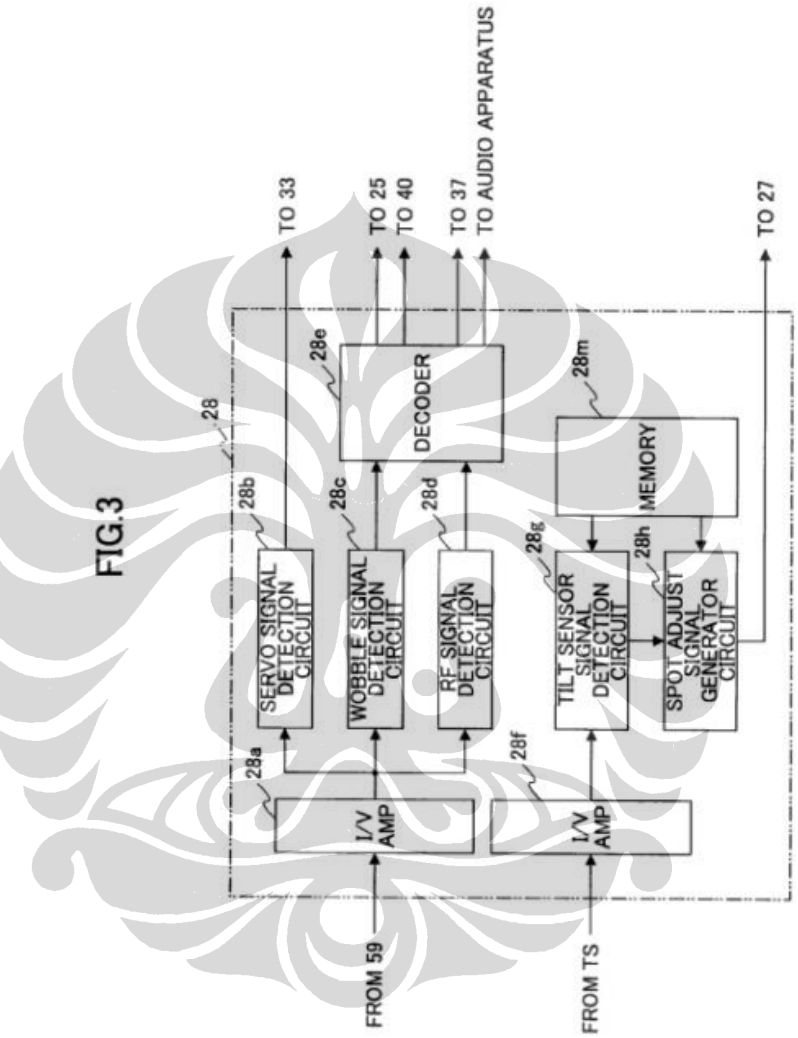


FIG.4

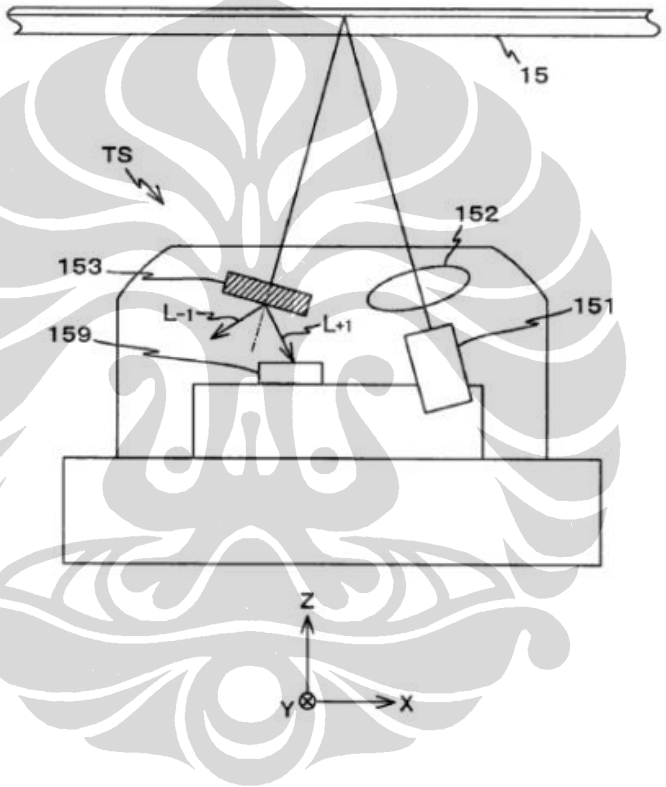


FIG.5

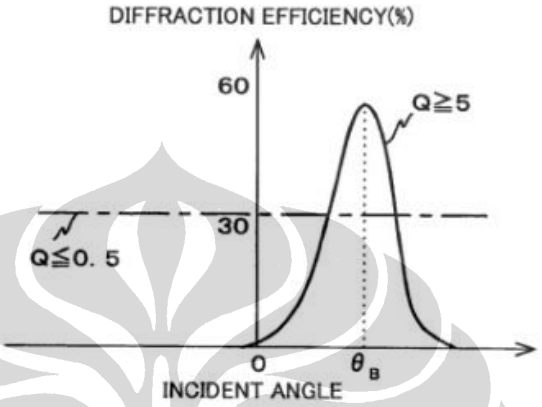


FIG.6

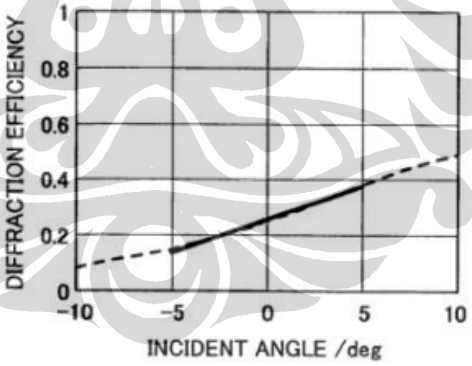


FIG.7A

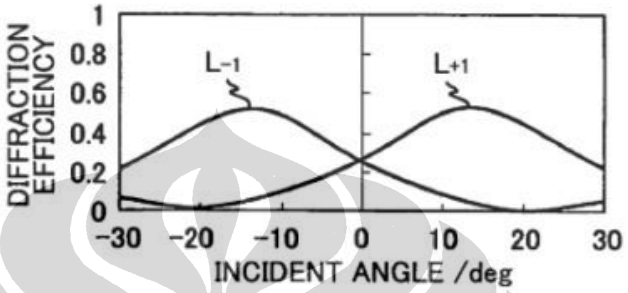


FIG.7B

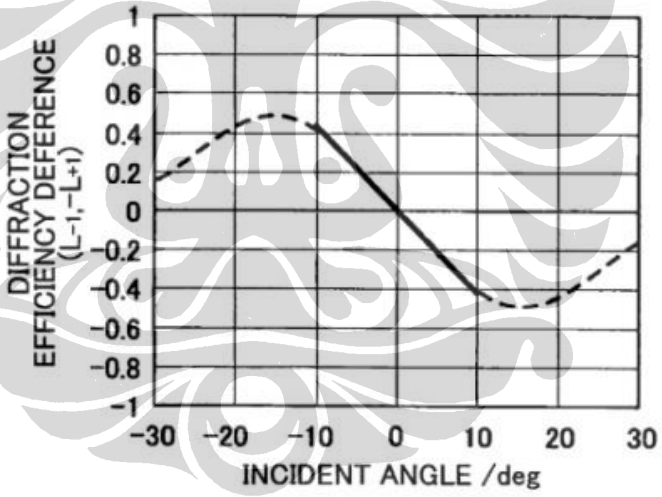


FIG.10A

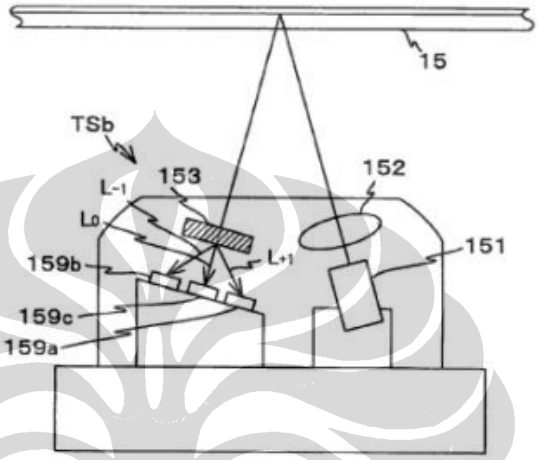


FIG.10B

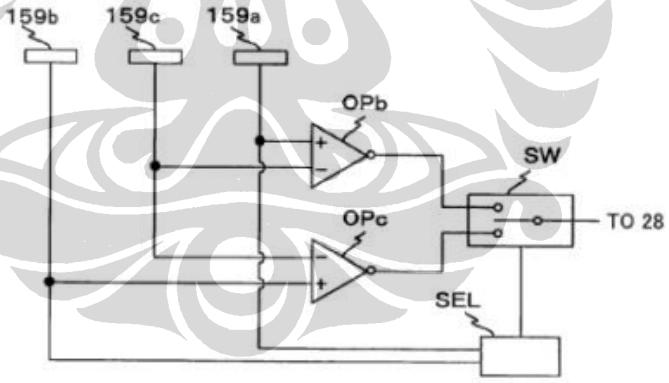


FIG.11

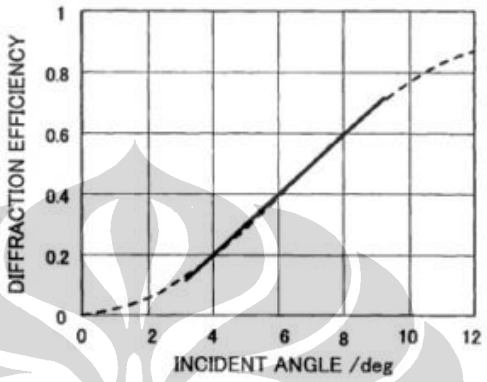


FIG.12

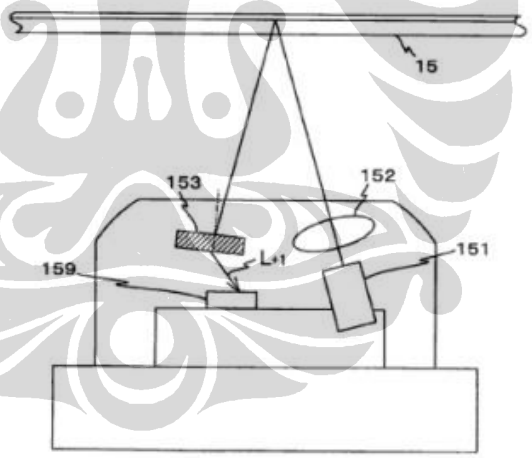


FIG.13

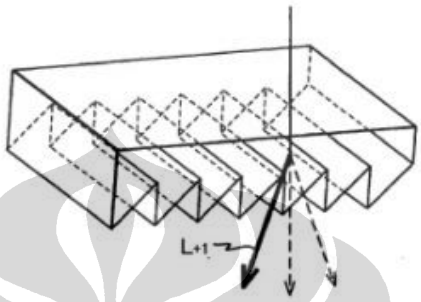


FIG.14

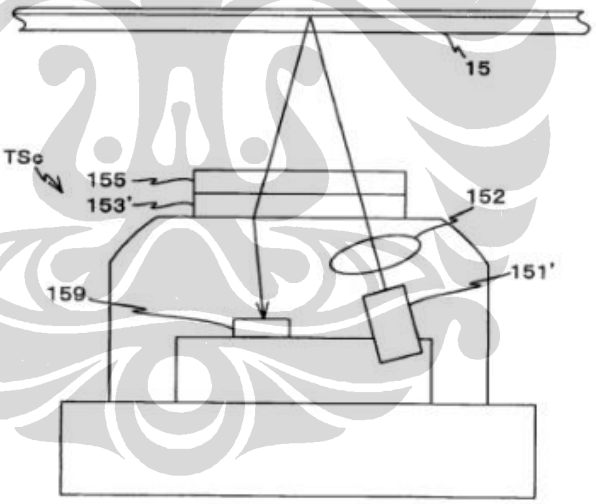


FIG.15A

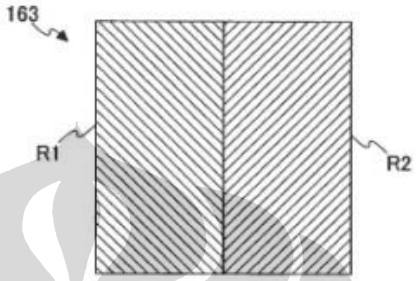


FIG.15B

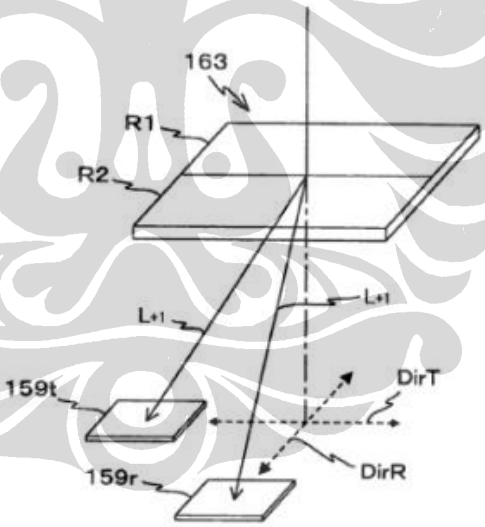


FIG.16

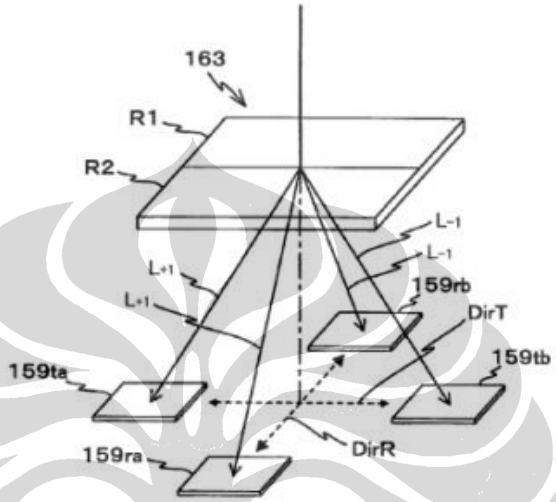


FIG.17

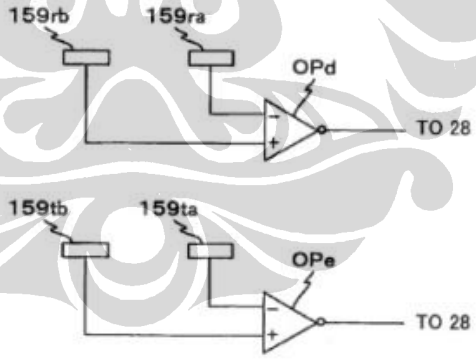


FIG.18

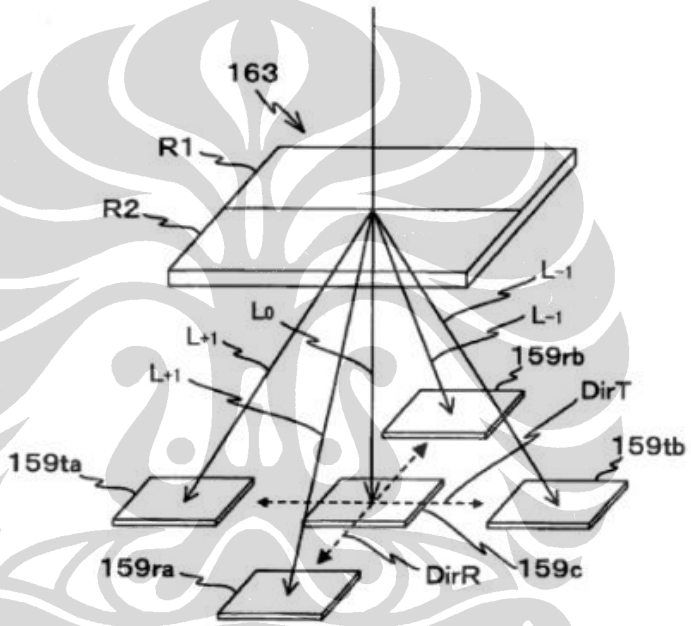


FIG.19

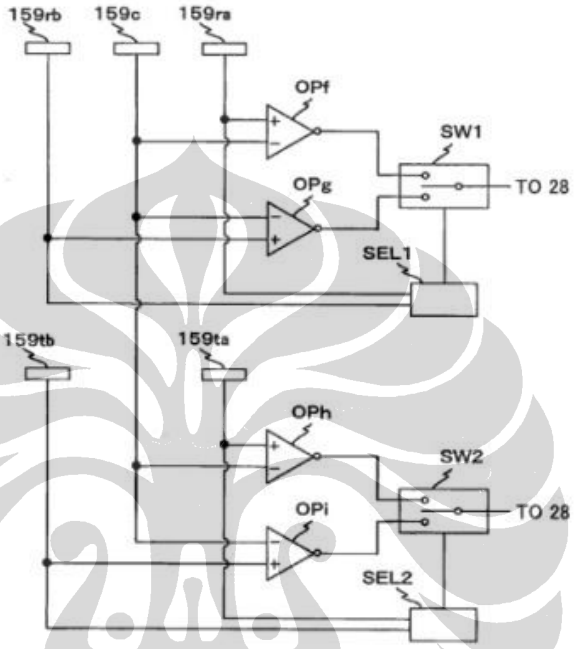


FIG.20

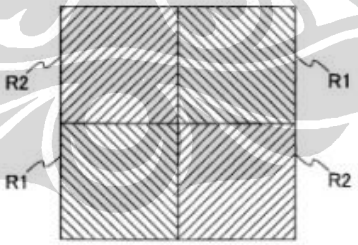


FIG.22

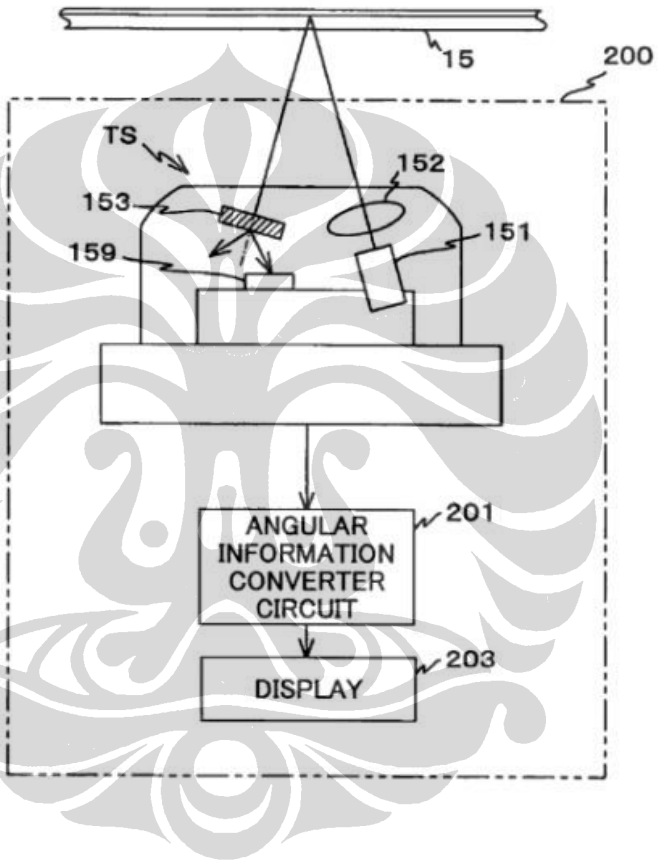


FIG.28

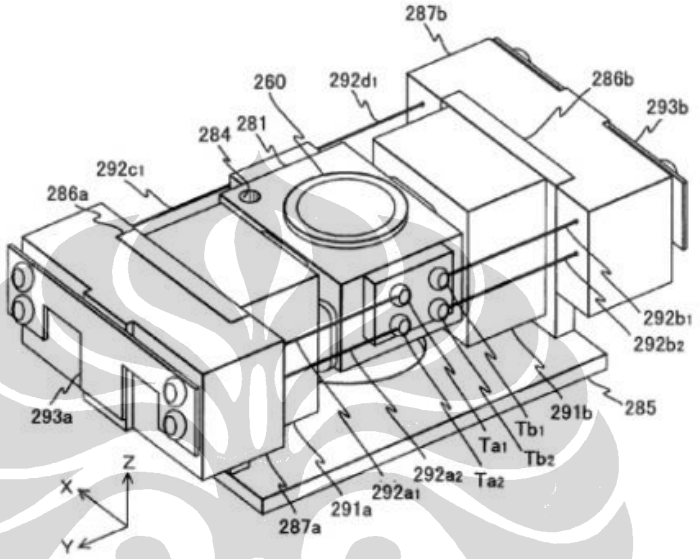


FIG.29

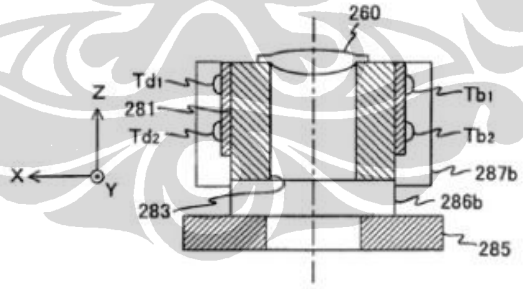


FIG.32

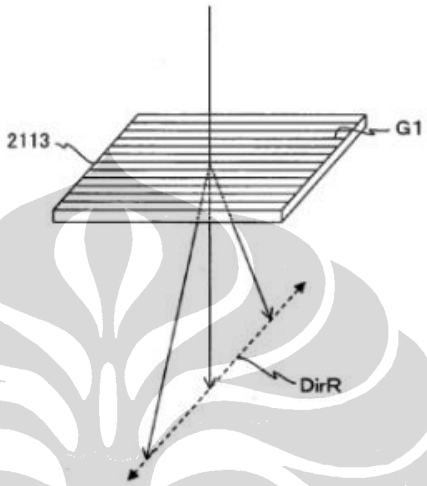


FIG.33

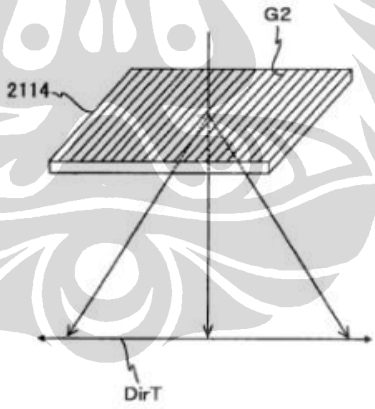


FIG.34

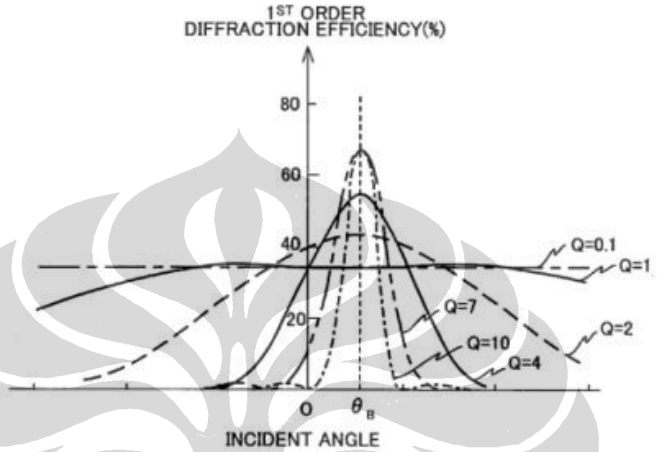


FIG.35

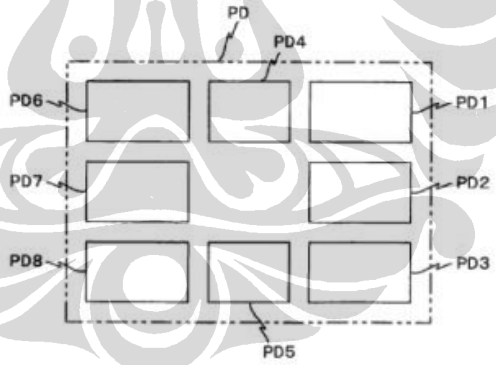


FIG.36

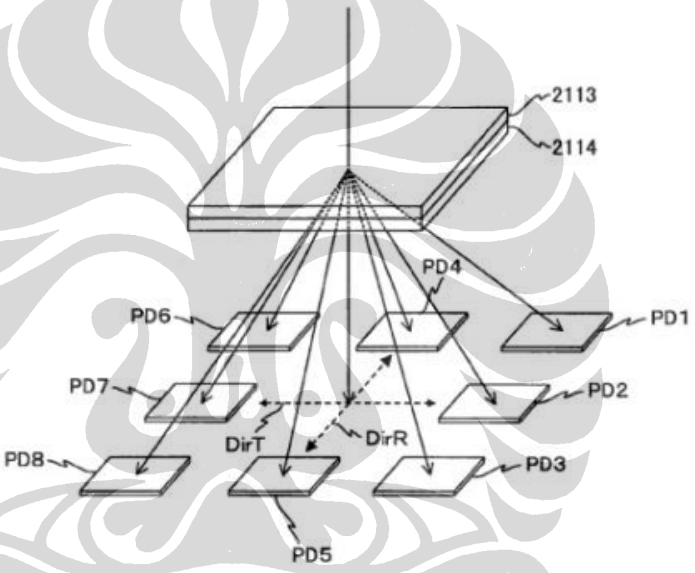


FIG.41

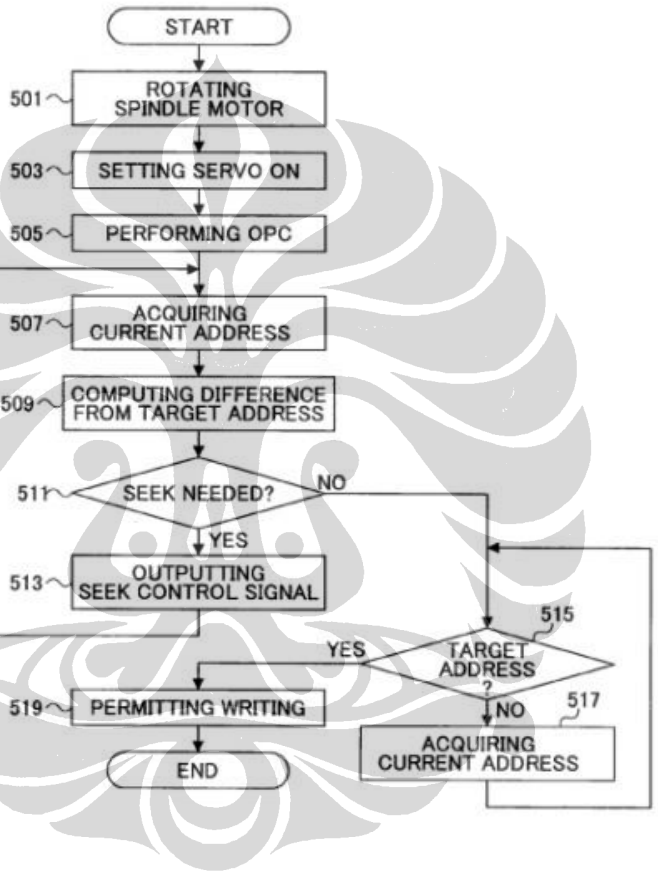


FIG.42

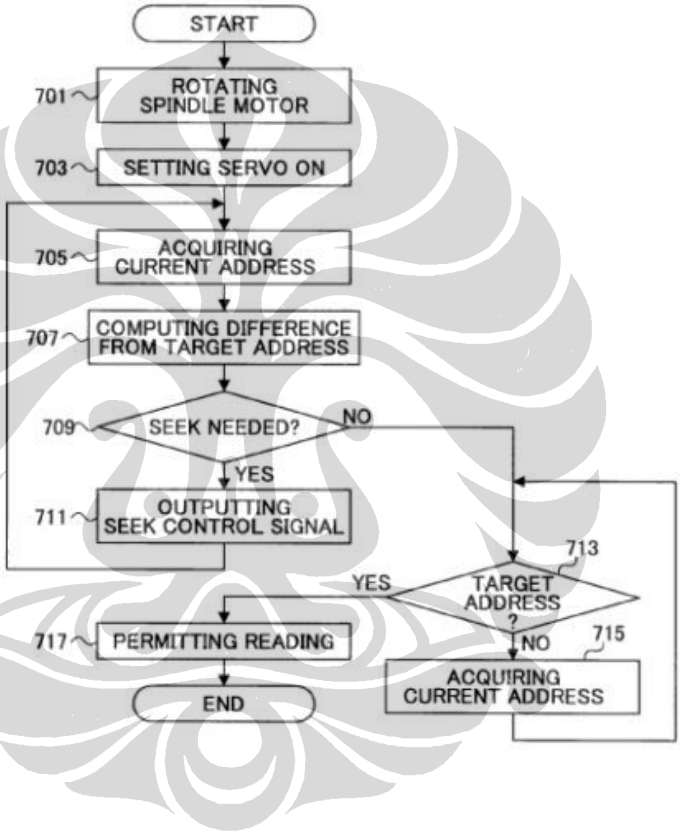


FIG.59

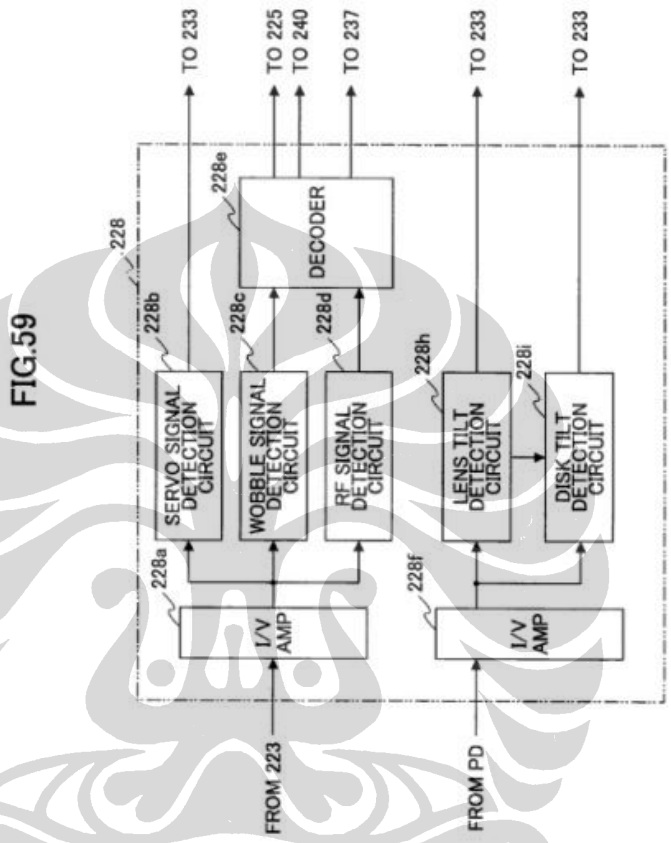


FIG.60

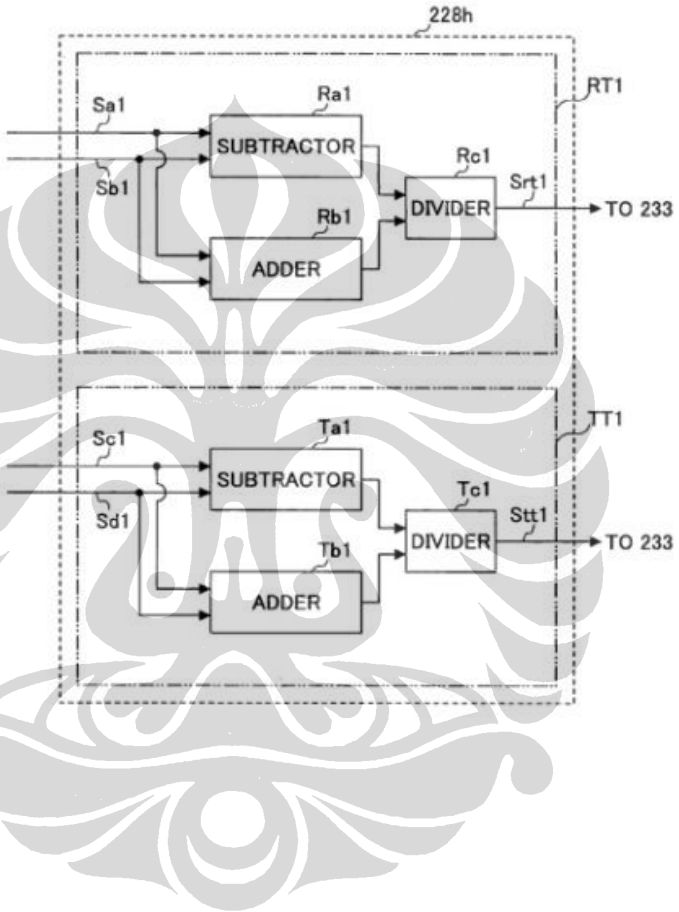


FIG.65

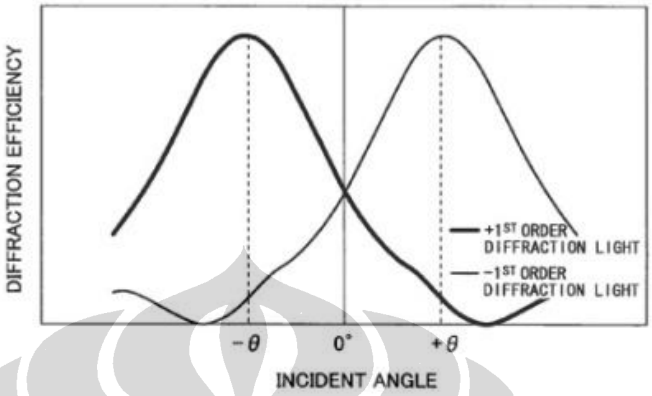


FIG.66

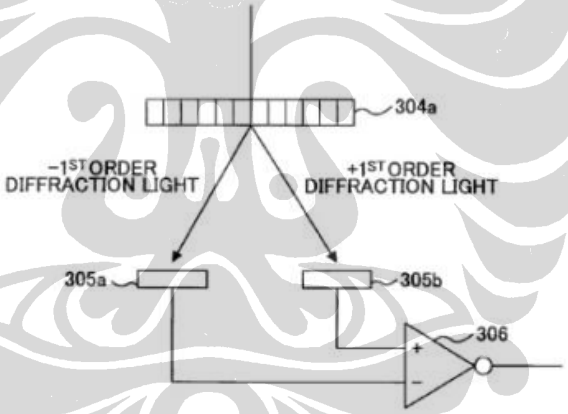


FIG.78

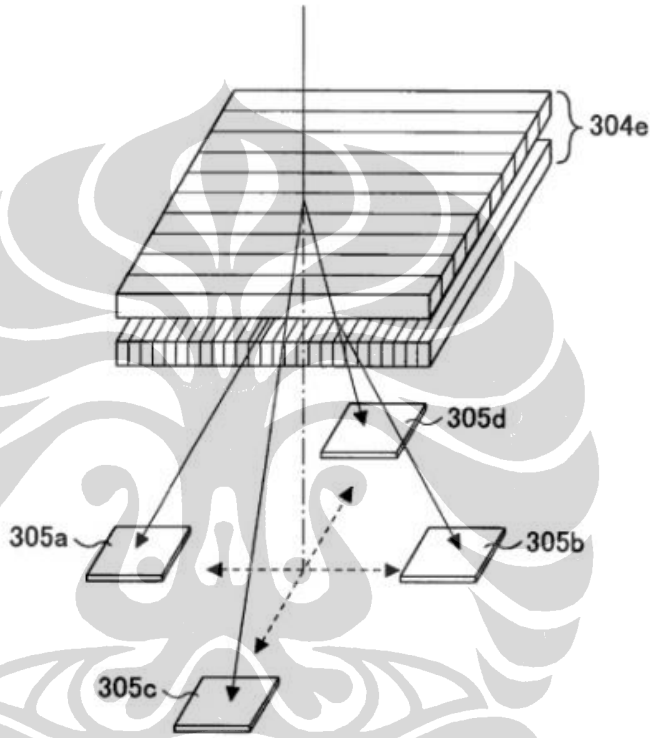


FIG.79

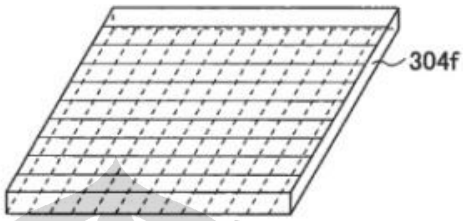


FIG.80

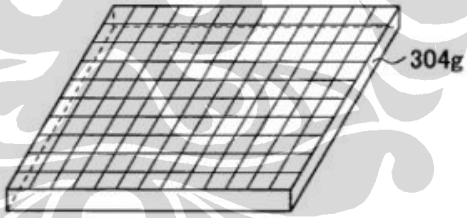


FIG.81

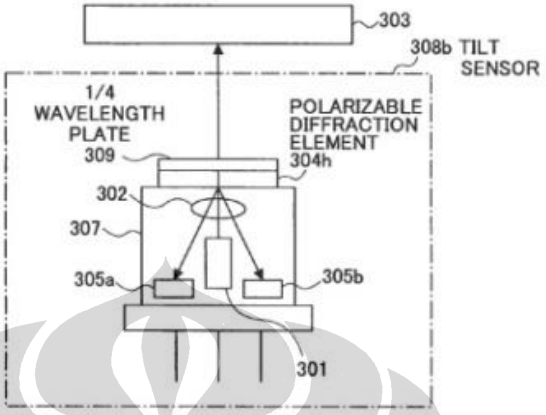


FIG.82