Birla Institute of Technology and Science

Pilani

A REPORT ON

Smart Lighting System

HEMANTH V ALLURI 2017A7PS1170P

RAMACHANDRAN SHANKAR 2017A7PS1171P

ROSHAN ROY 2017A7PS1172P

PRAVEEN RAVIRATHINAM 2017A7PS1174P

SUBMITTED TO Dr. P. Mishra

GROUP NO: 106

PROBLEM NO: 14

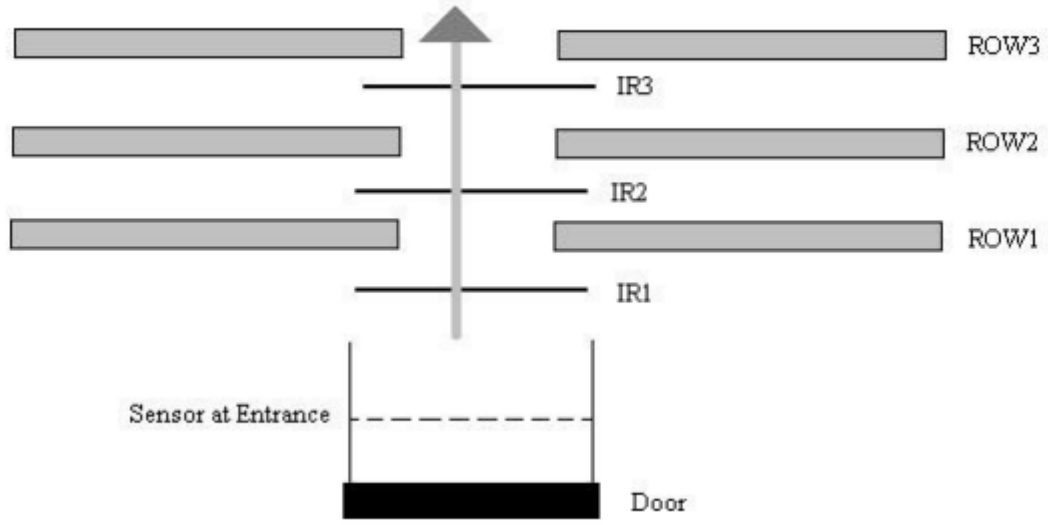
INDEX

PROBLEM STATEMENT

Problem statement 14:
System to be Designed: Smart Lighting System

Description: This is a lighting system for a conference room. On detection of a person the door is automatically opened/closed. As the seats get filled the light should be turned on. The rows are filled from row1 onwards. There are 3 lights per row. As each row begins to get filled the lights get turned on as each rows empties completely the light gets turned off. You can assume there are atleast 6 rows. The system should also display the number of people in the room on a LCD display.

System Details:



ASSUMPTIONS

1. There is only entry door to the conference room.
2. The same door is used for exit as well.
3. Only one person can enter at a time.
4. Only one person can exit at a time.
5. There is a gap of ~5 seconds between any two people's movements across the door.
6. There are 2 sensors on either side of the door.
7. Each row has 3 lights .
8. If the row has even a single person all the lights should still turn on.
9. People get seated from row 1 onwards and exit from only last occupied row. i.e. a person cannot exit from nth row if (n+1)th row has a person.
10. If there are people present in the nth row that means all n rows are lighted up.
11. Maximum capacity of the conference room is 30; 6 rows with each row capable of seating 5 people.

BRIEF USER GUIDE

This system design is meant to be extremely simple in terms of user interaction. There are only 2 inputs to the system. On each side of the sliding door to the conference room, there is a pressure plate the user will step over (modelled as push buttons in the model circuit). After stepping on the exterior pressure plate the sliding door will open within a 1-2 second interval. After the sliding door opens, the user will be allowed to step in and over the interior pressure plate (within about 2-3 seconds) which will confirm that the user entered. After a 5 second delay the door will then close, following which another user may enter or leave. Exiting follows a similar procedure but with the user stepping on the interior pressure plate first, then the exterior pressure plate.

LIST OF HARDWARE COMPONETS USED

CHIP / DEVICE	QUANTITY	USE
8086 Microprocessor	1	Central processing unit.
6116 – Ram 2K	2	Random access memory which contains [DS,SS]
2732 – ROM 4K	2	Read only memory which contains entire code [CS]
74LS373 OCTAL LATCH	1	For latching the address provided by the 8086.
74LS138	1	3-to-8 Decoder/Demultiplexer.
8255 PPI	2	PPI for connecting to peripheral components.
LED	18	Used for lighting up rows.
L293D	1	The driver for the bipolar stepper motor. [http://www.ti.com/lit/ds/symlink/l293.pdf]
7 SEG-COM-CATHODE	2	Used for displaying number of people in the room.
BIPOLAR STEPPER MOTOR	1	Used for controlling opening and closing of door.
4511 BCD to 7SEG DECODER	2	Decoders for the 7 segment displays.
NAND	1	Logic.
OR	9	Logic.
NOT	5	Logic.
AND	1	Logic.
100 ohm RESISTANCE	2	Logic.
PUSH BUTTONS	2	These are the two sensors on either side of the door

ADDRESS MAPPING OF DEVICES

MEMORY MAPPING:

The system uses 4KB of RAM and 8KB of ROM. RAM consists of two 2K chips and ROM consists of 4K chips. They are organized into odd and even bank to facilitate both byte and word size data transfers.

Read Only Memory: Starting Address: 00000h, Ending Address: 01FFFh

Random Access Memory: Starting Address: 02000h, Ending Address: 02FFFh

CHIP	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ROM :FROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ROM :TO	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RAM :FROM	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
RAM :TO	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

I/O MAPPING:

PPI1 – Pressure Plates (Push Buttons), Bipolar Stepper Motor Control, 7SEG Display

PORTA - 0000h

PORTB - 0002h

PORTC - 0004h

CONTROL REGISTER - 0006h

PPI2 – Lighting Array

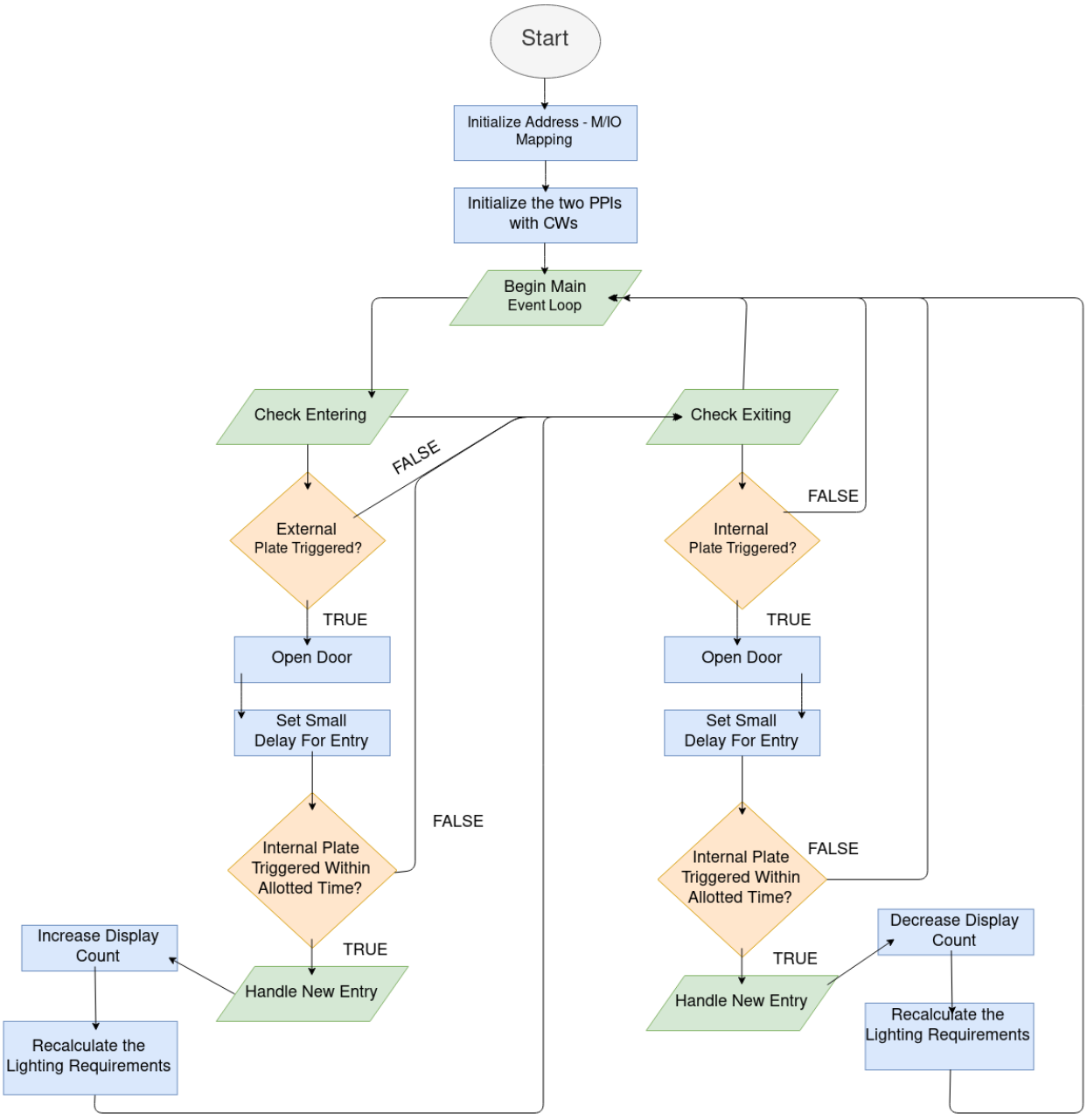
PORTA - 0001h

PORTB - 0003h

PORTC - 0005h

CONTROL REGISTER - 0007h

FLOWCHART


Start
Initialize Address - M/IO Mapping
Initialize the two PPIs with CWs
Begin Main Event Loop
Check Entering
Check Exiting
External Plate Triggered?
FALSE
TRUE
Open Door
Set Small Delay For Entry
Internal Plate Triggered Within Allotted Time?
FALSE
TRUE
Handle New Entry
Increase Display Count
Recalculate the Lighting Requirements
Internal Plate Triggered?
FALSE
TRUE
Open Door
Set Small Delay For Entry
Internal Plate Triggered Within Allotted Time?
FALSE
TRUE
Handle New Entry
Decrease Display Count
Recalculate the Lighting Requirements

CODE

```
.MODEL TINY
.DATA
   PORTA1 equ 0000h
   PORTB1 equ 0002h
   PORTC1 equ 0004h
   CWREG1 equ 0006h

   PORTA2 equ 0001h
   PORTB2 equ 0003h
   PORTC2 equ 0005h
   CWREG2 equ 0007h


.CODE
.STARTUP

   init:
    ORG  0000h

    ; This register will serve as internal memory and store the number of people in the room.
    MOV  dl, 00h

    ; Set up the first PPI to read from Port A and write to Port B and Port C.
    MOV  al, 10010000b
    OUT  CWREG1, al
```

```
  ; Set up the second PPI to write to Port C [each output line is a row of 3 lights].
  MOV  al, 10010010b
  OUT  CWREG2, al


; "main" serves as our event loop and the functions check_entry and check_exit will handle the rest.
main:
  CALL check_entry
  CALL check_exit
  JMP  main


delay PROC NEAR
; A small timed delay. The caller should set CX before calling.
  delay1:
    NOP
    DEC  cx
    JNE  delay1
    RET
delay ENDP

update_display PROC NEAR
; Update the 2 7SEGs to display the number of people in the room.
    MOV  bl, 10d
    MOV  ax, dx
    DIV  bl ; al: quotient, ah: remainder
    MOV  bl, ah
    SHL  ax, 4
    ADD  al, bl
```

```
   OUT  PORTC1, al
   RET
update_display ENDP

incr_cnt PROC NEAR
; Increase the count value and update the display.
   INC  dl;
   CALL update_display;
   CALL light_rows
   RET
incr_cnt ENDP


decr_cnt PROC NEAR
; Decrease the count value and update the display.
   DEC  dl;
   CALL update_display;
   CALL light_rows
   RET
decr_cnt ENDP


open_door PROC NEAR
; Open the door by turning the stepper motor to 180deg.
   MOV  al, 00000001b
   OUT  PORTB1, al
   RET
open_door ENDP
```

```
close_door PROC NEAR
; Close the door by turning the stepper motor to 0deg.
    MOV  al, 00000010b
    OUT  PORTB1, al
    RET
close_door ENDP



check_entry PROC NEAR
    ; Check to see if the external pressure sensor has been triggered.
    IN   al, PORTA1
    CMP  al, 00000001b
    JNE  check_entry4

    ; Open the door once the external pressure sensor has been triggered.
    CALL open_door
    MOV  cx, 0FFFFh
    CALL delay

    ; Check to see if the internal pressure sensor has been triggered. Provide a small window of time for entry.
    MOV  cx, 0FFFFh
  check_entry1:
    IN   al, PORTA1
    CMP  al, 00000010b
    JE   check_entry2
    DEC  cx
    JNZ  check_entry1
```

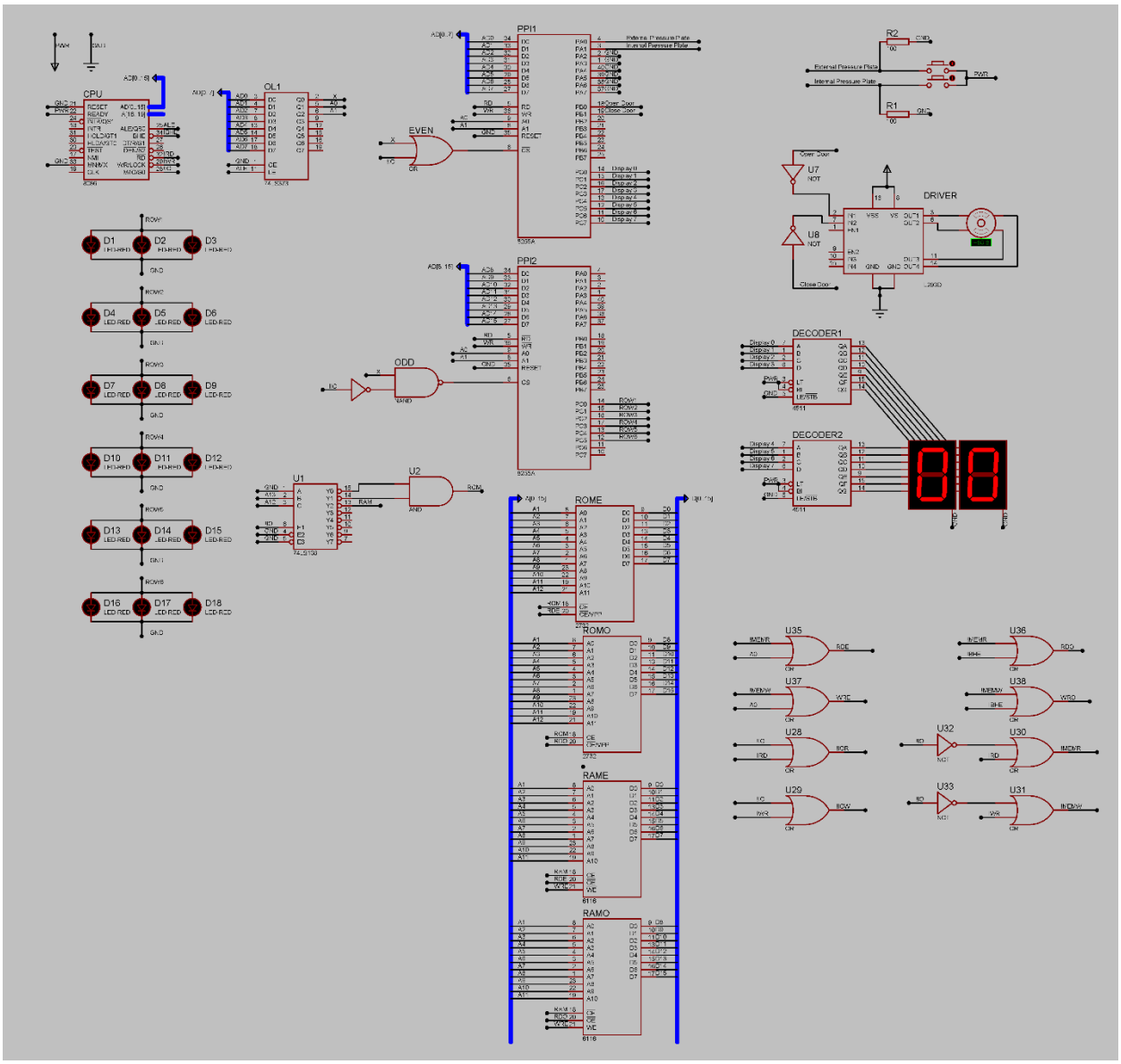
```
    JMP  check_entry3

  check_entry2:
    ; If the person entered then the count should be incremented.
    CALL incr_cnt

  check_entry3:
    ; Close the door.
    CALL close_door
  check_entry4:
    RET
 check_entry ENDP


 check_exit PROC NEAR
    ; Check to see if the external pressure sensor has been triggered.
    IN   al, PORTA1
    CMP  al, 00000010b
    JNE  check_exit4

    ; Open the door once the external pressure sensor has been triggered.
    CALL open_door
    MOV  cx, 0FFFFh
    CALL delay

    ; Check to see if the internal pressure sensor has been triggered. Provide a small window of time for
entry.
    MOV  cx, 0FFFFh
  check_exit1:
```

```
    IN   al, PORTA1
    CMP  al, 00000001b
    JE   check_exit2
    DEC  cx
    JNZ  check_exit1
    JMP  check_exit3

  check_exit2:
    ; If the person exited then the count should be decremented.
    CALL decr_cnt

  check_exit3:
    ; Close the door.
    CALL close_door
  check_exit4:
    RET
check_exit ENDP


light_rows PROC NEAR
    ; Check to see how many people are in the room (using the value in dl),
    ; then accordingly determine the number of rows that need to be lit up.
    ; start filling from ROW1 onwards.
    ; Our assumption is that the rows seat 5 people each.
    ; Thus the total capacity of the conference room is 5*6 = 30 people.
    MOV  al, 00d

    CMP  dl, 25d
    JLE  light_rows1
```

```
  INC  al
light_rows1:
  SHL  al, 1 ; Moving this up by one line would slightly improve speed (heuristic).

  CMP  dl, 20d
  JLE  light_rows2
  INC  al
light_rows2:
  SHL  al, 1

  CMP  dl, 15d
  JLE  light_rows3
  INC  al
light_rows3:
  SHL  al, 1

  CMP  dl, 10d
  JLE  light_rows4
  INC  al
light_rows4:
  SHL  al, 1

  CMP  dl, 05d
  JLE  light_rows5
  INC  al
light_rows5:
  SHL  al, 1

  CMP  dl, 00d
```

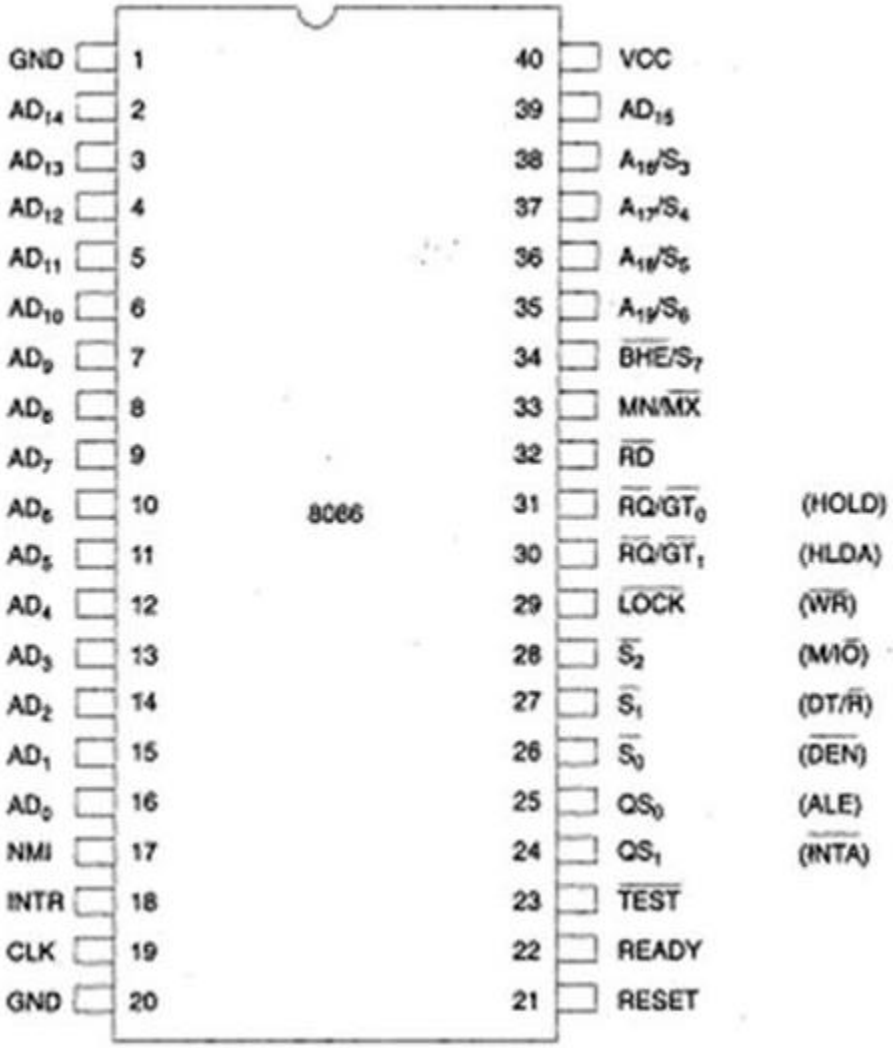
```
        JE  light_rows6
        INC  al

      light_rows6:
        OUT  PORTC2, al

        RET
    light_rows ENDP

END
```

CIRCUIT DIAGRAM



EXTRA REFERENCES

8086 PINOUT DIAGRAM:



STEPPER MOTOR:

The data sheet and other information for the driver for the bipolar stepper motor can be found in the following link:

http://www.ti.com/lit/ds/symlink/l293.pdf