Mohammad Ali Jinnah University

Department of Computer Systems Engineering

Two Way Traffic Light Using Arduino + 8255

MORTAZA HASSANI	FA19-BECE-0007
M.AZEEMULLAH KHAN	FA19-BECE-2001
SIDHARTH TANWANI	FA19-BECE-0011

Project report submitted to Sir M. Sufiyan

in partial fulfilment of (Microprocessor's Interfacing) lab project in the semester of (SPRING21, 4th)

Contents

Abstract

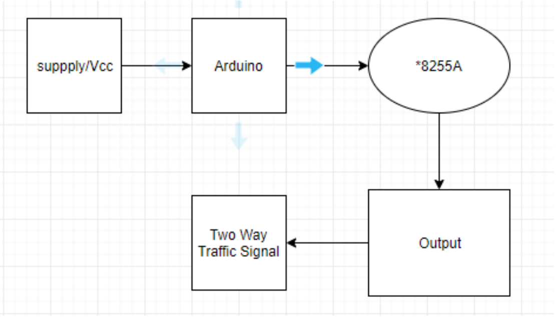
Purpose of making this is to implement the idea of giving traffic lights signals using Arduino and interfacing with 8255 plus 8086 interfacing 8255.

Introduction

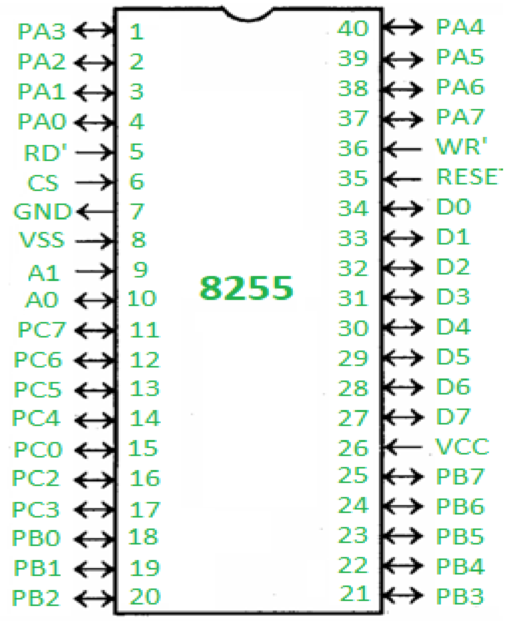
Signals offer maximum control at intersections. They relay messages of both what to do and what not to do. The primary function of any traffic signal is to assign right-of-way to conflicting movements of traffic at an intersection. This is done by permitting conflicting streams of traffic to share the same intersection by means of time separation. By alternately assigning right of way to various traffic movements, signals provide for the orderly movement of conflicting flows. They may interrupt extremely heavy flows to permit the crossing of minor movements that could not otherwise move safely through an intersection.

Hardware & Design

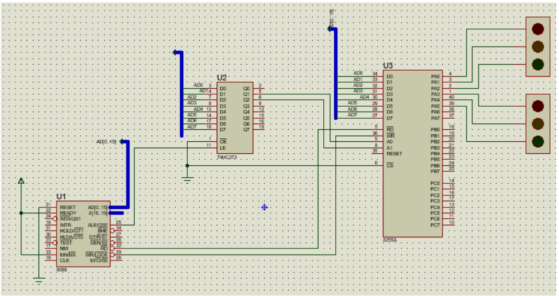
Block Diagram



8255A :The 8255A is a general purpose programmable I/O device designed to transfer the data from I/O to interrupt I/O under certain conditions as required. It can be used with almost any microprocessor.



Schematic with 8086



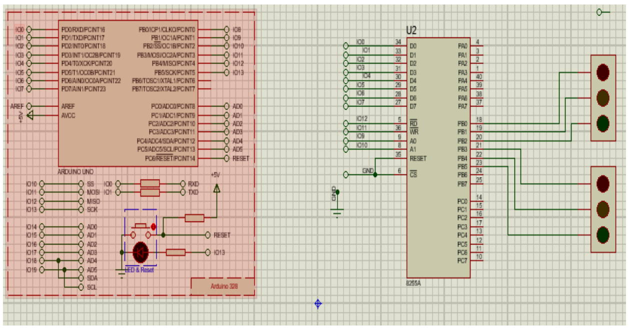
8086 CODE

```
DATA SEGMENT
 PORTA EQU 00H
 PORTB EQU 02H
 PORTC EQU 04H
 PORT_CON EQU 06H
DATA ENDS
CODE SEGMENT
 MOV AX,DATA
 MOV DS, AX


ORG 0000H
START:


 MOV DX, PORT_CON
 MOV AL, 10000010B; port C (output), port A (output) in
mode 0 and PORT B (INPUT) in mode 0
 OUT DX, AL
 mov bx,0
 mov cx,101
l1:
```

```
MOV AL,  00100001B
MOV DX, PORTA
OUT DX,AL
loop l1
 mov cx,10
jmp l2
l2:
 MOV AL, 00010010B
MOV DX, PORTA
OUT DX,AL
loop l2
mov cx,89
jmp l3
 l3:
 MOV AL, 00001100B
MOV DX, PORTA
OUT DX,AL
loop l3
mov cx,69
l4:
 MOV AL,  00010010B
MOV DX, PORTA
OUT DX,AL
loop l4
JMP START
CODE ENDS
END
```

Schematic for Arduino along with 8255 and Traffic lights

Schematic with Arduin & 8255



Purpose

Purpose is not using Arduino our actual objective is to use 8255A

From 8255A we can increase digital ports of Arduino

Working

We are using digital pins of Arduino as output i.e pin 0 to 7. The other pine we are using for read n write able and disable and 2 pins we are using for selecting port i.e a0,a1.

We are giving control signals(control word) from Arduino to 8255A then we are selecting values of a0 and a1 on which port we want output. Here we are using port b so all our LEDs are on port B then we are making a function in which we are obtaining output 1 from our desired port number . then we are using delay function for each led to turn on after specific time and then we are running it same in loop to do it continuously.

Components & Program

Arduino

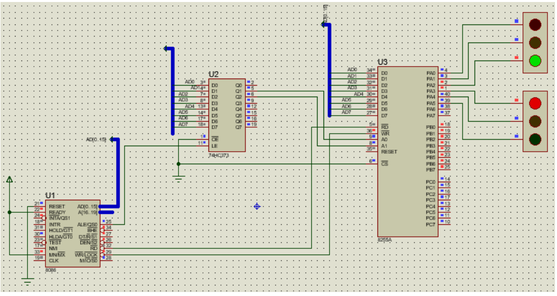
Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino Source code

```
const int a0=9;
const int a1=10;
const int WR=11;
const int RD=12;

void setup()
{
  pinMode(0,OUTPUT);
  pinMode(1,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(a0,OUTPUT);
  pinMode(a1,OUTPUT);
  pinMode(WR,OUTPUT);
  pinMode(RD,OUTPUT);
  digitalWrite(RD,1);    //Disabling READ
  digitalWrite(WR,1);   //Disabling WRITE

  digitalWrite(a0,1);   //Selecting Control
Register
  digitalWrite(a1,1);   //Selecting Control
Register
  digitalWrite(7,1);   //
  digitalWrite(6,0);   //
  digitalWrite(5,0);   //
  digitalWrite(4,0);   //
  digitalWrite(3,0);   //
  digitalWrite(2,0);   //
  digitalWrite(1,0);   //
  digitalWrite(0,0);   //
  digitalWrite(WR,0);
  delay(500);
  digitalWrite(WR,1);
}
void portb0on(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(0,1);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
 void portb0off(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(0,0);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
  void portb1on(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(1,1);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
void portb1off(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(1,0);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
   void portb2on(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(2,1);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
   void portb2off(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(2,0);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
   void portb3on(){
  digitalWrite(a0,0);
  digitalWrite(a1,0);
    digitalWrite(3,1);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
 void portb3off(){
  digitalWrite(a0,0);
  digitalWrite(a1,0);
    digitalWrite(3,0);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
   void portb4on(){
  digitalWrite(a0,1);
  digitalWrite(a1,0);
    digitalWrite(4,1);
    digitalWrite(WR,0);
    digitalWrite(1,0);
   digitalWrite(WR,1);
   }
  void portb4off(){
```

```
 digitalWrite(a0,1);
 digitalWrite(a1,0);
  digitalWrite(4,0);
  digitalWrite(WR,0);
  digitalWrite(1,0);
 digitalWrite(WR,1);
 }
 void portb5on(){
 digitalWrite(a0,1);
 digitalWrite(a1,0);
  digitalWrite(5,1);
  digitalWrite(WR,0);
digitalWrite(1,0);
 digitalWrite(WR,1);
 }
 void portb5off(){
 digitalWrite(a0,1);
 digitalWrite(a1,0);
  digitalWrite(5,0);
  digitalWrite(WR,0);
  digitalWrite(1,0);
 digitalWrite(WR,1);
 }
 void loop(){
portb0on();
portb5on();
delay(5000);
portb0off();
portb5off();
delay(1000);
portb4on();
portb1on();
delay(500);
portb4off();
portb1off();
delay(1000);
portb3on();
portb2on();

delay(5000);
portb3off();
portb2off();

delay(1000);
portb4on();
portb1on();
delay(500);
portb4off();
portb1off();
}
```

Result



For this configuration at every time from the two Traffic lights one of the LEDs is on indication of allowing on route to pass and other one to stop. If 1st Traffic light is green then 2nd traffic light at the same time is red and vice versa. Before changing state of red/green the yellow light indication turns on as alert and then state changes.

Difficulties in implementation

There were many difficulties while we were making this project

- Ports were not giving the right output
- Delay function was not being implemented properly while giving control signals on Arduino
- Some of the ports were even not working when I placed LEDs on those ports.
- Shorting the ports didn’t led us to configure it out.

Conclusion

The result aimed achieved successfully. For further expansion of this project can use more traffic lights and set manual control also with 7 segments in order to display time remaining for each signal.