Unit - III

Programmable Peripheral Interface

Topics

- Keypad Interfacing
- LED Display Interfacing
- Stepper Motor Control
- Traffic Control Interface

Keypad Interfacing

Key Scanning Procedure

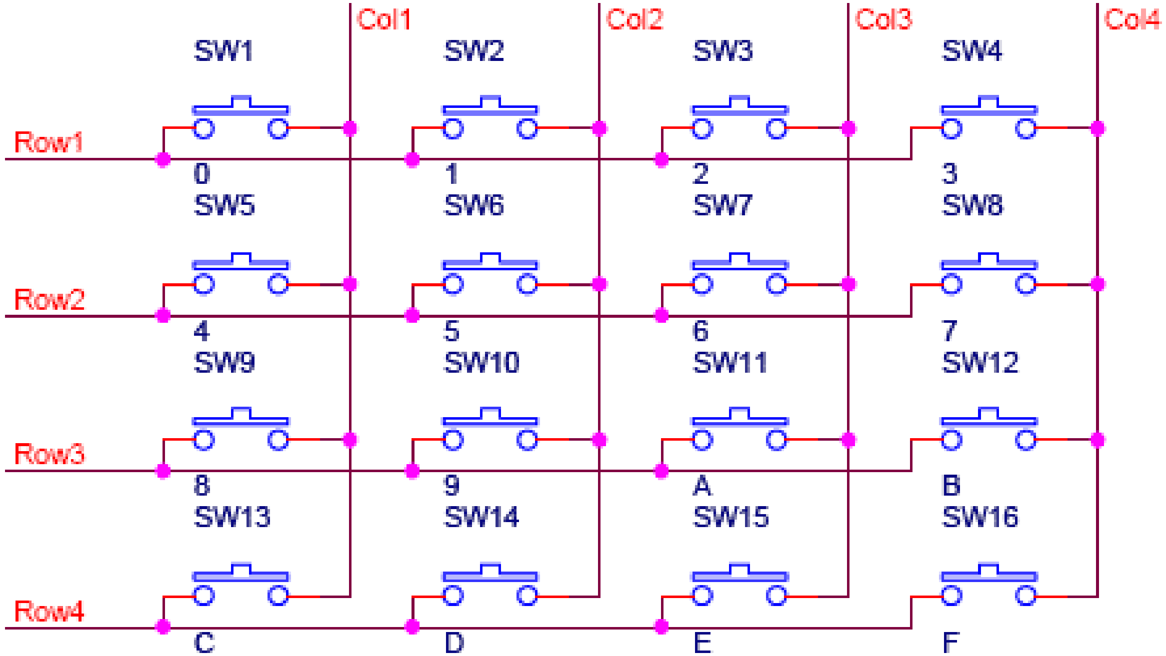
Getting meaningful data from a keyboard, it requires the following three major tasks:

- Detect a keypress.
- Debounce the keypress.
- Encode the keypress

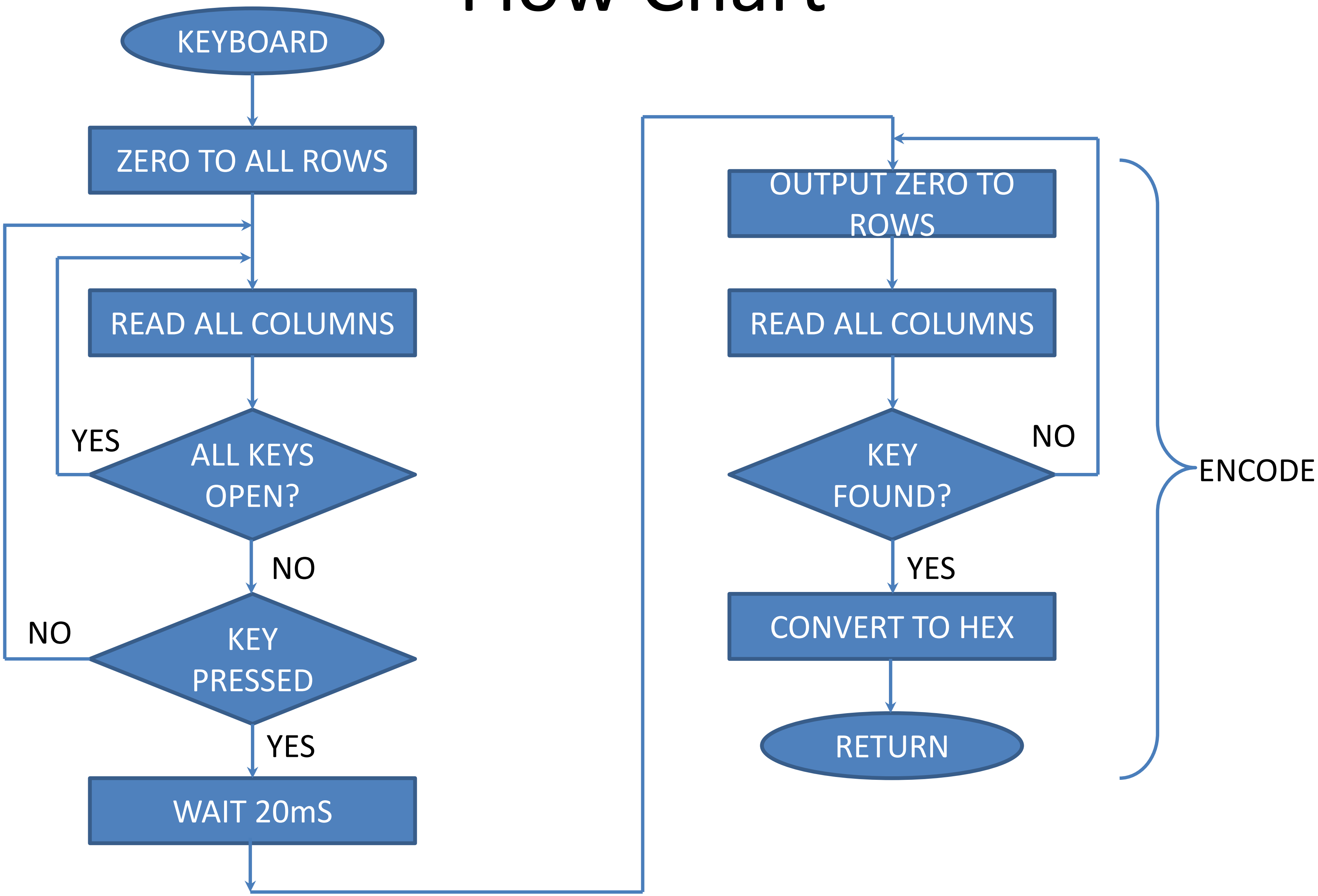
Three tasks can be done with hardware, software, or a combination of two, depending on the application.

Circuit Connections

- Matrix of keys interface



Flow Chart



KEYBOARD
ZERO TO ALL ROWS
READ ALL COLUMNS
YES
ALL KEYS OPEN?
NO
NO
KEY PRESSED
YES
WAIT 20mS
OUTPUT ZERO TO ROWS
READ ALL COLUMNS
NO
KEY FOUND?
YES
CONVERT TO HEX
RETURN
ENCODE

Idea to identify key pressed

If R1=1, R2=0, R3=0, R4=0

C1=1, C2=0, C3=0, C4=0 => SW1 Pressed
C1=0, C2=1, C3=0, C4=0 => SW2 Pressed
C1=0, C2=0, C3=1, C4=0 => SW3 Pressed
C1=0, C2=0, C3=0, C4=1 => SW4 Pressed

If R1=0, R2=1, R3=0, R4=0

C1=1, C2=0, C3=0, C4=0 => SW5 Pressed
C1=0, C2=1, C3=0, C4=0 => SW6 Pressed
C1=0, C2=0, C3=1, C4=0 => SW7 Pressed
C1=0, C2=0, C3=0, C4=1 => SW8 Pressed

If R1=0, R2=0, R3=1, R4=0

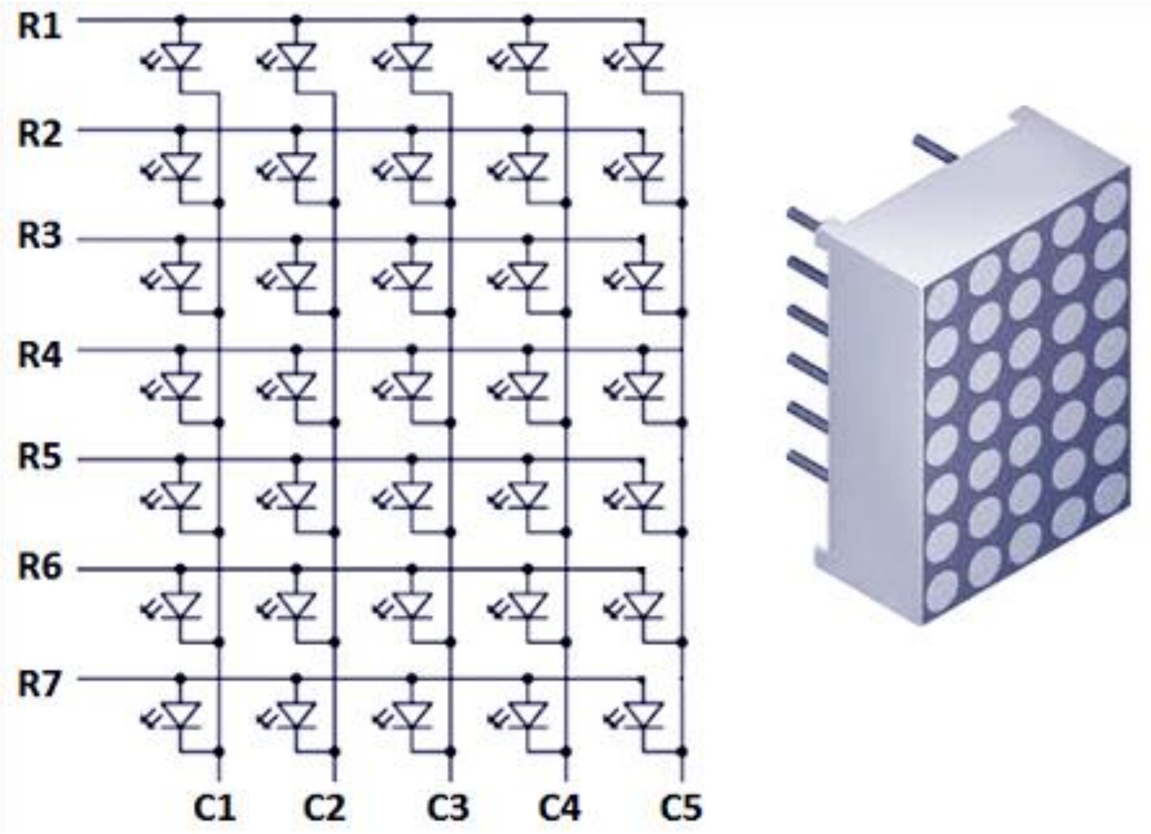
C1=1, C2=0, C3=0, C4=0 => SW9 Pressed
C1=0, C2=1, C3=0, C4=0 => SW10 Pressed
C1=0, C2=0, C3=1, C4=0 => SW11 Pressed
C1=0, C2=0, C3=0, C4=1 => SW12 Pressed

If R1=0, R2=0, R3=0, R4=1

C1=1, C2=0, C3=0, C4=0 => SW13 Pressed
C1=0, C2=1, C3=0, C4=0 => SW14 Pressed
C1=0, C2=0, C3=1, C4=0 => SW15 Pressed
C1=0, C2=0, C3=0, C4=1 => SW16 Pressed

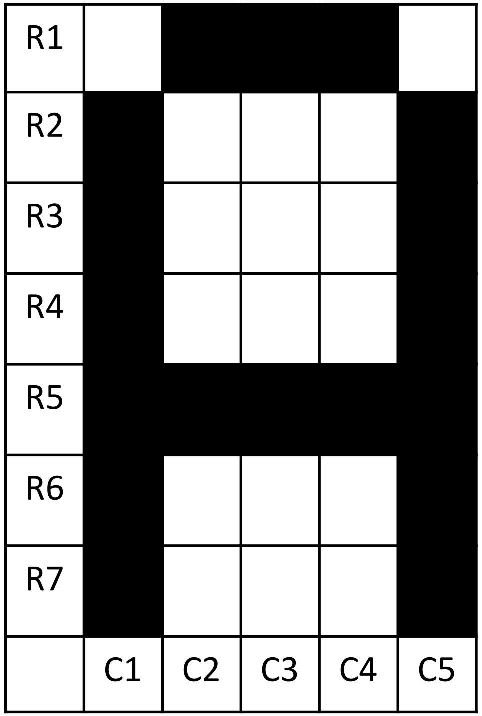
LED Display Interfacing

LED Display Interfacing



To Display Character 'A'

R1	R2	R3	R4	R5	R6	R7	C1	C2	C3	C4	C5
0	1	1	1	1	1	1	0	1	1	1	1
Delay 1mSec											
1	0	0	0	1	0	0	1	0	1	1	1
Delay 1mSec											
1	0	0	0	1	0	0	1	1	0	1	1
Delay 1mSec											
1	0	0	0	1	0	0	1	1	1	0	1
Delay 1mSec											
0	1	1	1	1	1	1	1	1	1	1	0
Delay 1mSec											



Stepper Motor Control

Types by Construction

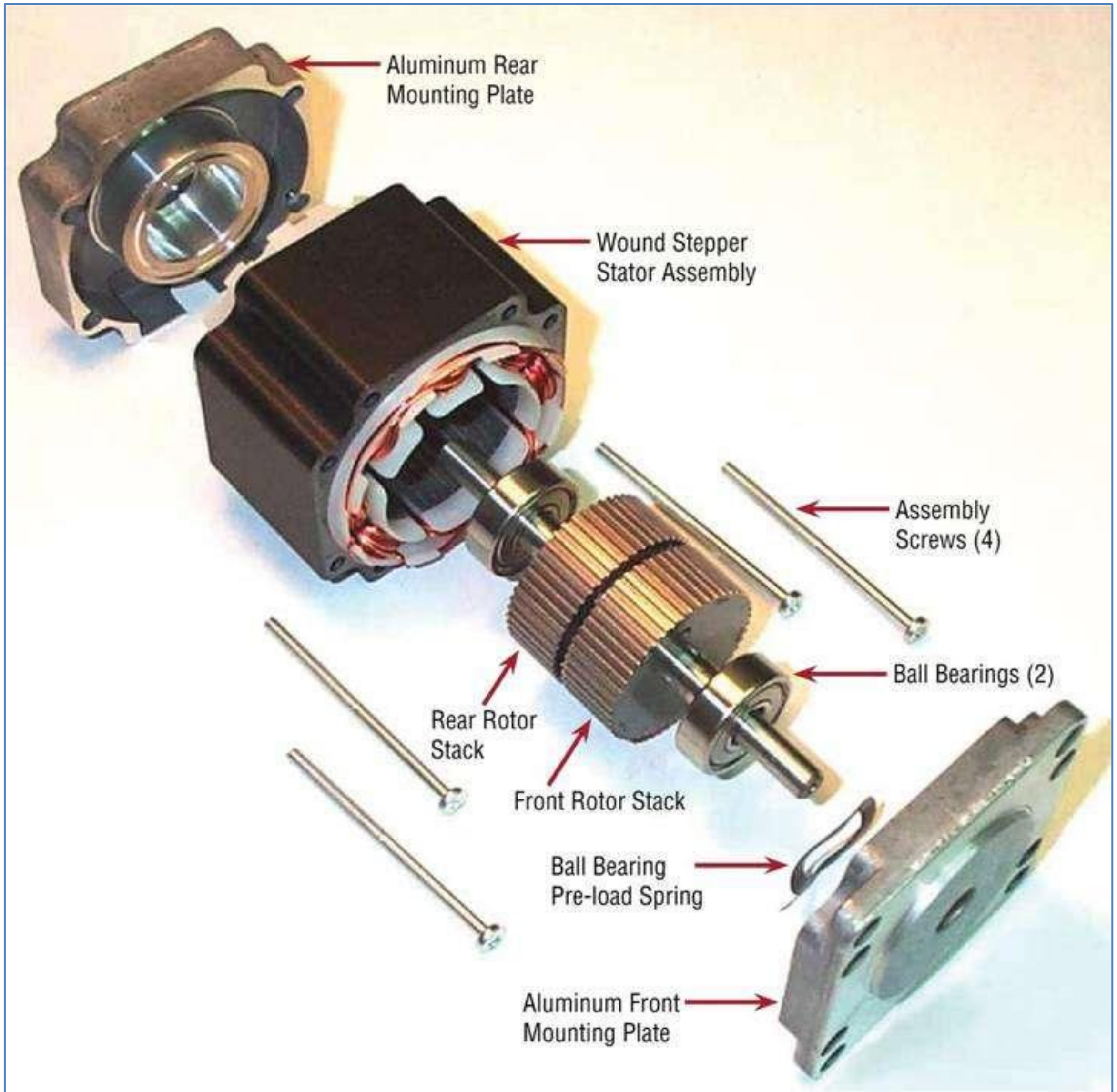
By construction there are 3 different types of (rotors)stepper motors:

Permanent magnet stepper,

Variable reluctance stepper and

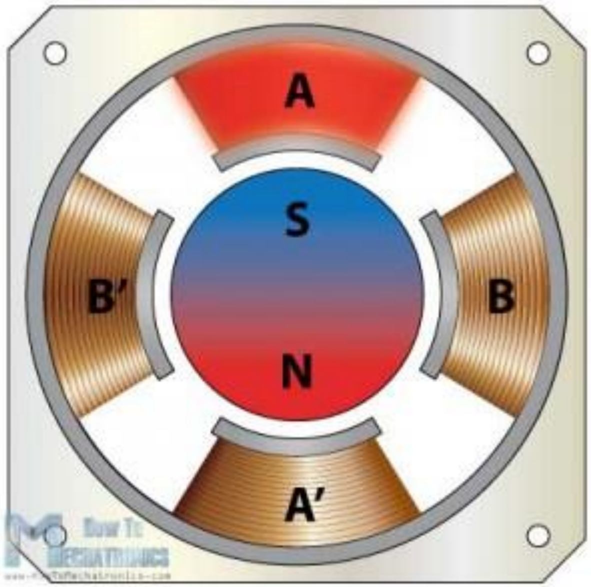
Hybrid synchronous stepper motor.

Stepper Motor Parts



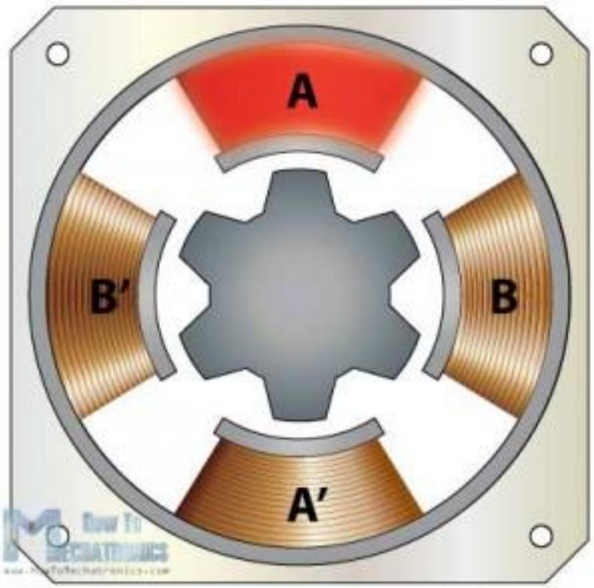
Permanent magnet stepper

The **Permanent Magnet** stepper has a **permanent magnet rotor** which is driven by the stators windings. They create opposite polarity poles compared to the poles of the rotor which propels the rotor.



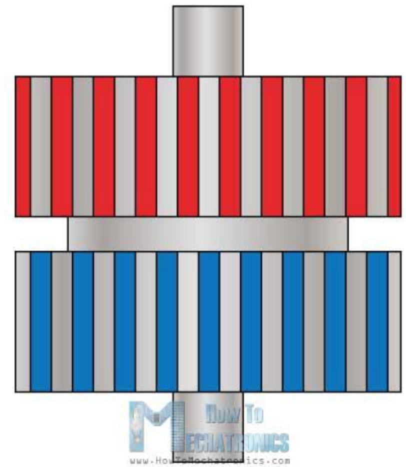
Variable Reluctant stepper

The next type, the **Variable Reluctant** stepper motor uses a **non-magnetizes soft iron rotor**. The **rotor has teeth** that are offset from the stator and as we active the windings in a particular order the rotor moves respectively so that it has minimum gab between the stator and the teeth of the rotor

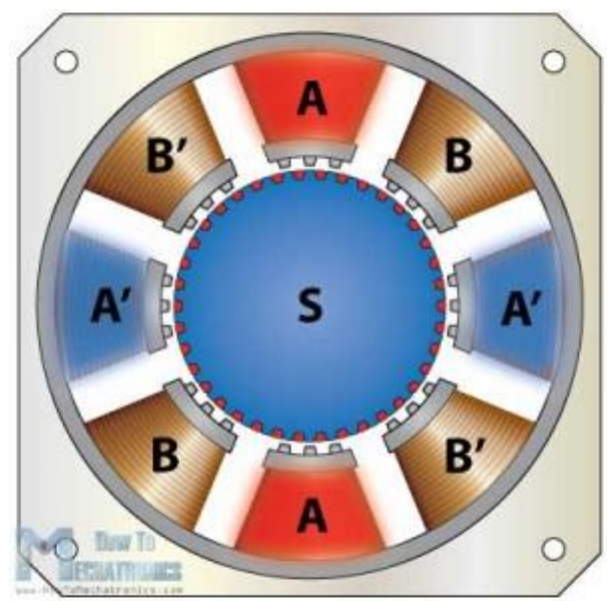


Hybrid Synchronous Stepper

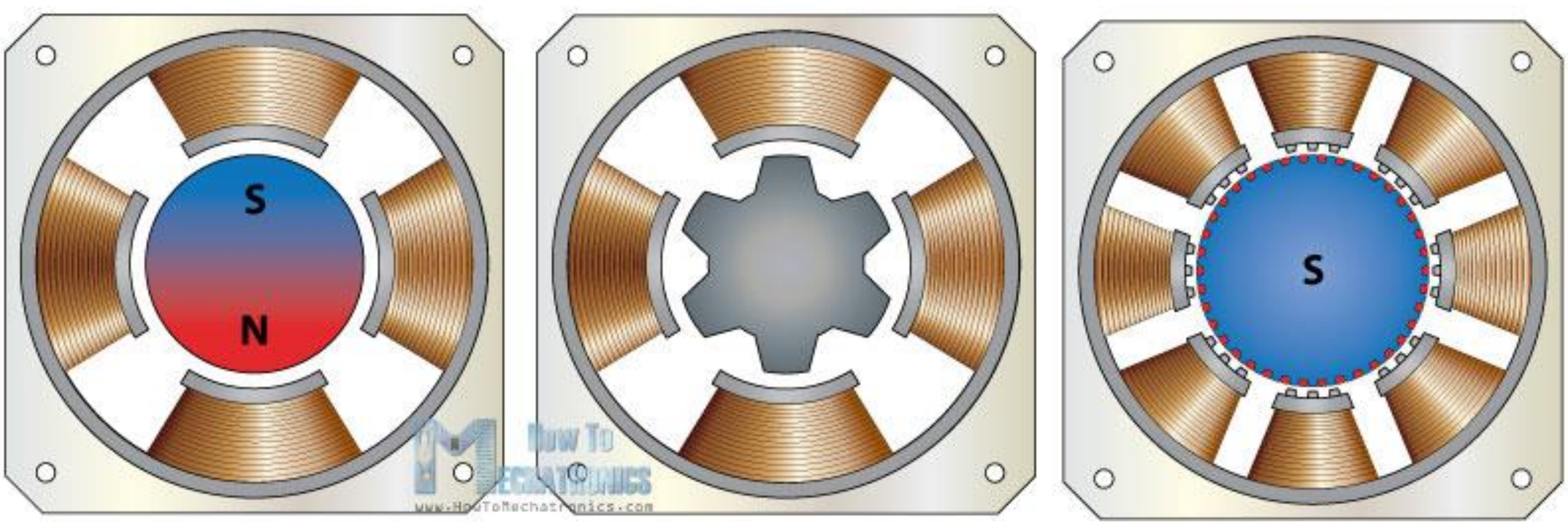
The **Hybrid Synchronous** motor is combinations of the previous two steppers. It has **permanent magnet toothed rotor and also a toothed stator**. The rotor has two sections, which are opposite in polarity and their teeth are offset as shown here.



This is a front view of a commonly used hybrid stepper motor which has 8 poles on the stator that are activated by 2 windings, A and B. So if we activate the winding A, we will magnetize 4 poles of which two of them will have South polarity and two of them North polarity.



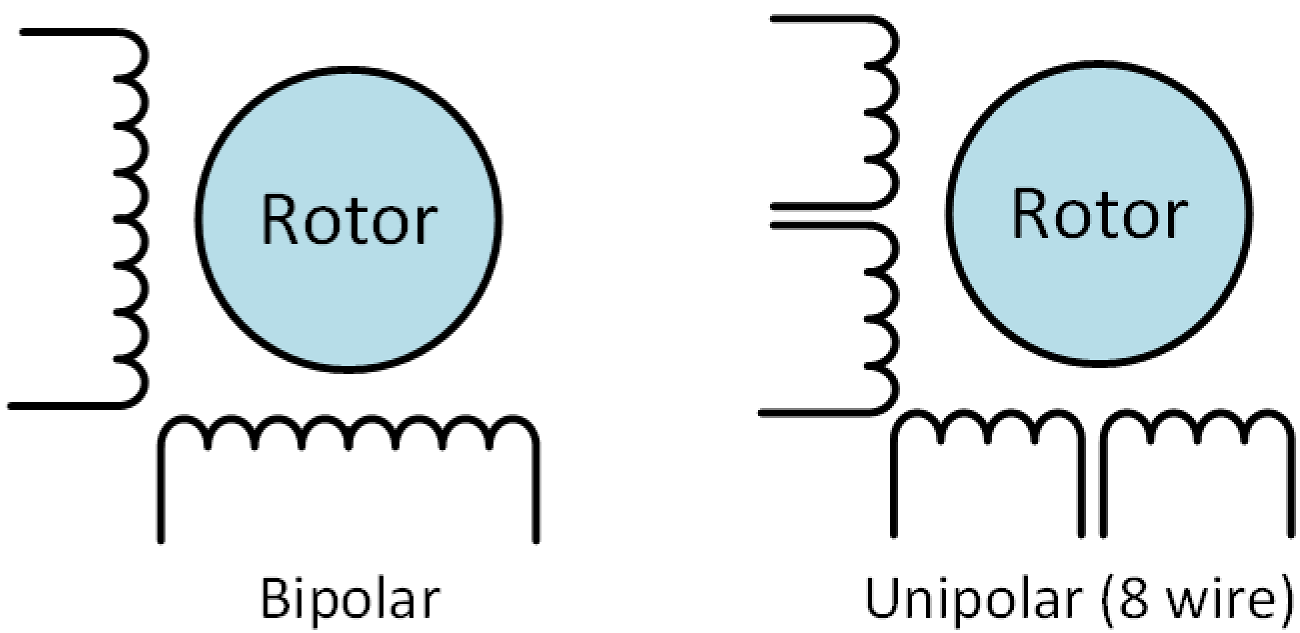
Comparision



Types by Coils and Phases

Coils and Phases

A stepper motor may have any number of coils. But these are connected in groups called "phases". All the coils in a phase are energized together.



Unipolar vs. Bipolar

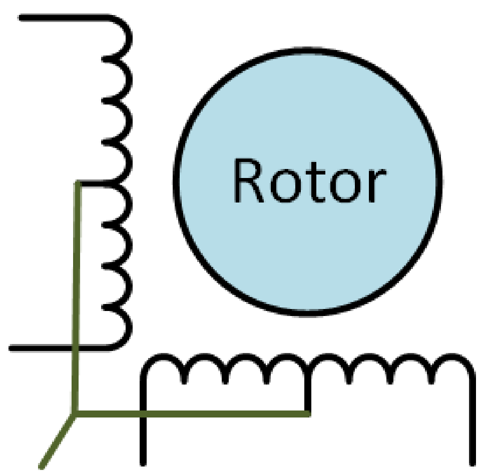
Unipolar drivers, always energize the phases in the same way. **One lead, the "common" lead, will always be negative. The other lead will always be positive.** Unipolar drivers can be implemented with simple transistor circuitry. The disadvantage is that there is less available torque because only half of the coils can be energized at a time.

Bipolar drivers use H-bridge circuitry to actually reverse the current flow through the phases. By energizing the phases with alternating the polarity, all the coils can be put to work turning the motor.

A two phase bipolar motor has 2 groups of coils. A 4 phase unipolar motor has 4 groups of coils. A 2-phase bipolar motor will have 4 wires - 2 for each phase. Some motors come with flexible wiring that allows you to run the motor as either bipolar or unipolar.

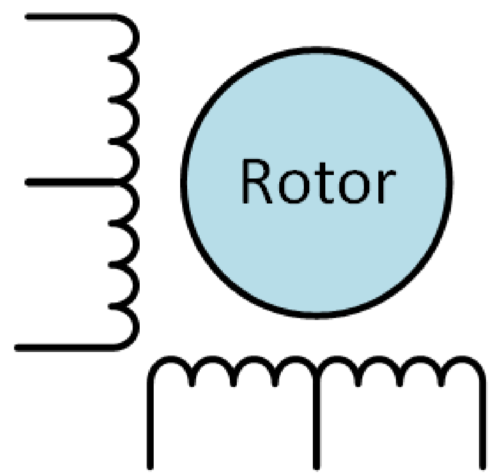
Types by Coils and Phases

5-Wire Motor
This style is common in smaller unipolar motors. All of the common coil wires are tied together internally and brought out as a 5th wire. This motor can only be driven as a unipolar motor.



Unipolar (5 wire)

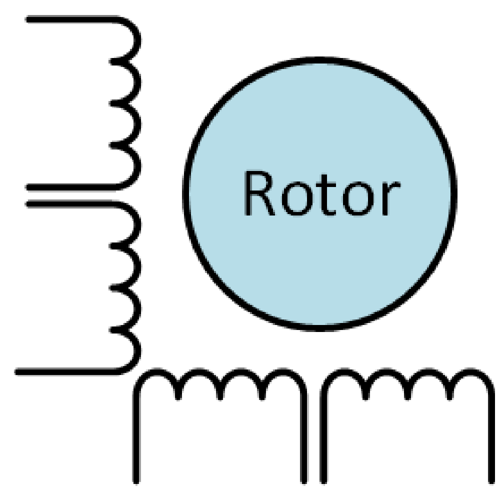
6-Wire Motor
This motor only joins the common wires of 2 paired phases. These two wires can be joined to create a 5-wire unipolar motor.
Or you just can ignore them and treat it like a bipolar motor!



Unipolar (6 wire)

Types by Coils and Phases

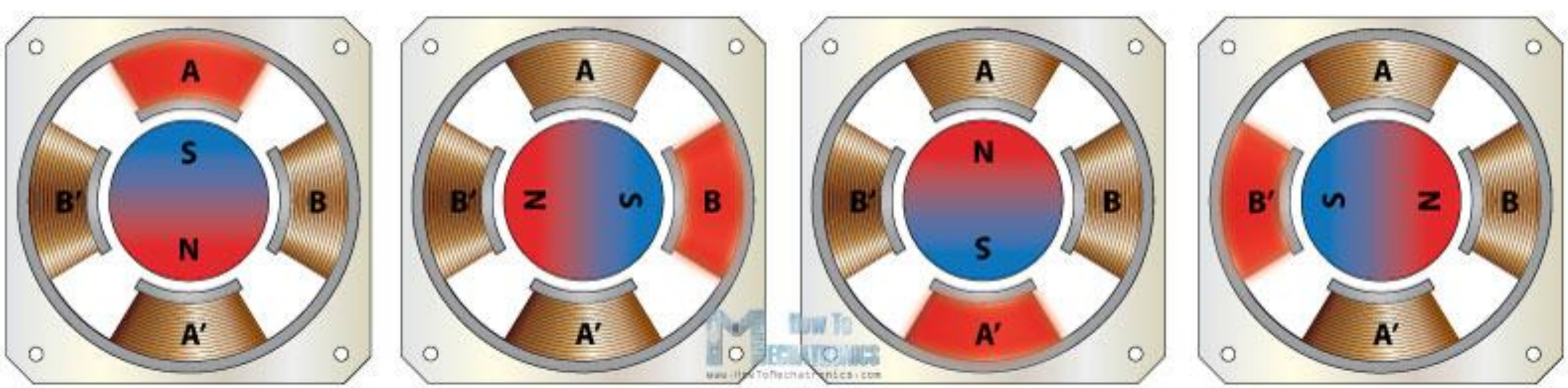
8-Wire Motor
The 8-wire unipolar is the most versatile motor of all. It can be driven in several ways:
4-phase unipolar - All the common wires are connected together - just like a 5-wire motor.
2-phase series bipolar - The phases are connected in series - just like a 6-wire motor.
2-phase parallel bipolar - The phases are connected in parallel. This results in half the resistance and inductance - but requires twice the current to drive. The advantage of this wiring is higher torque and top speed.



Unipolar (8 wire)

Working Principle

Stepper motor is a brushless DC motor that rotates in steps. This is very useful because it can be precisely positioned without any feedback sensor, which represents an open-loop controller. The stepper motor consists of a rotor that is generally a permanent magnet and it is surrounded by the windings of the stator. As we activate the windings step by step in a particular order and let a current flow through them they will magnetize the stator and make electromagnetic poles respectively that will cause propulsion to the motor.



Driving Modes

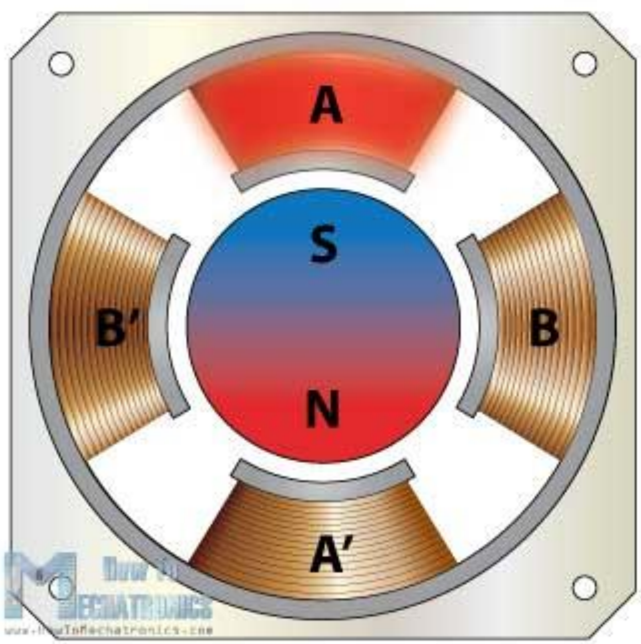
Four types of driving modes

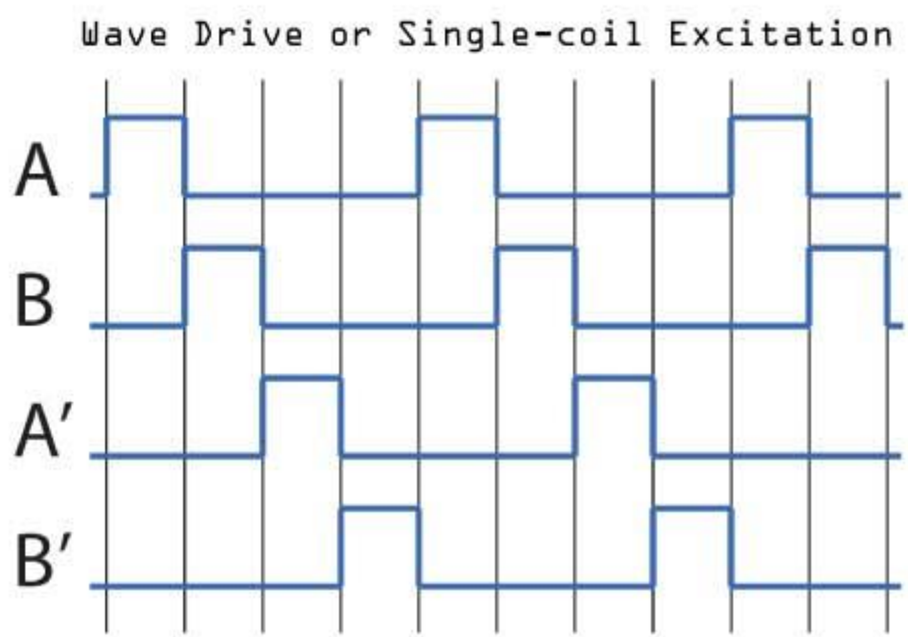
1. Wave drive or Single coil excitation
2. Full Step Drive
3. Half Step Drive
4. Microstepping

Wave drive or Single coil excitation

The first one is the Wave Drive or Single-Coil Excitation. In this mode we active just one coil at a time which means that for this example of motor with 4 coils, the rotor will make full cycle in 4 steps.

A	B	A’	B’
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

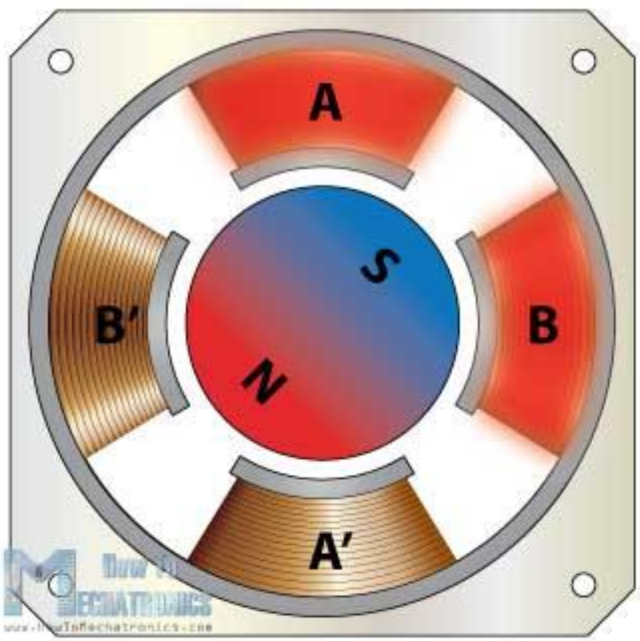


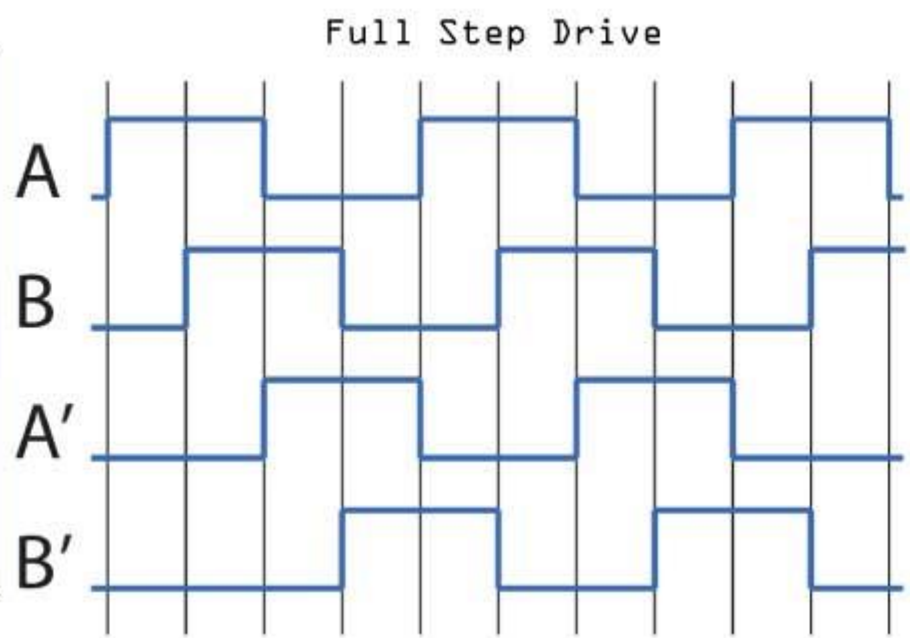


Full Step Drive

Next is the Full step drive mode which provides much higher torque output because we always have 2 active coils at a given time. However this doesn't improve the resolution of the stepper and again the rotor will make a full cycle in 4 steps.

A	B	A'	B'
1	0	0	1
1	1	0	0
0	1	1	0
0	0	1	1

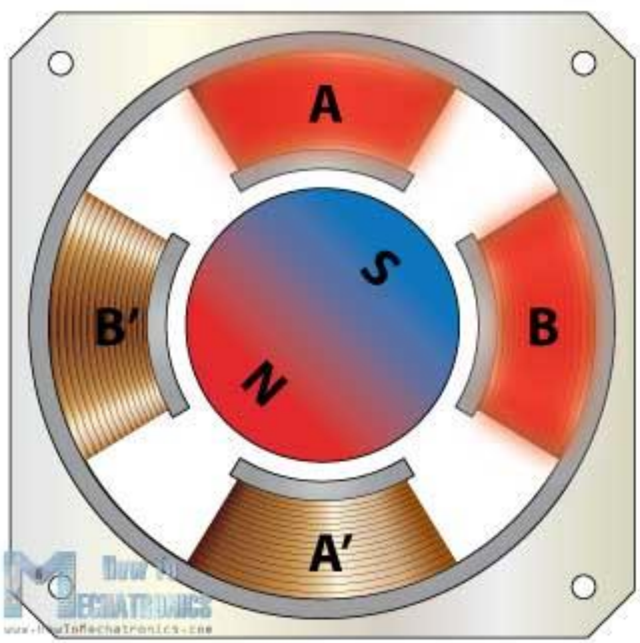


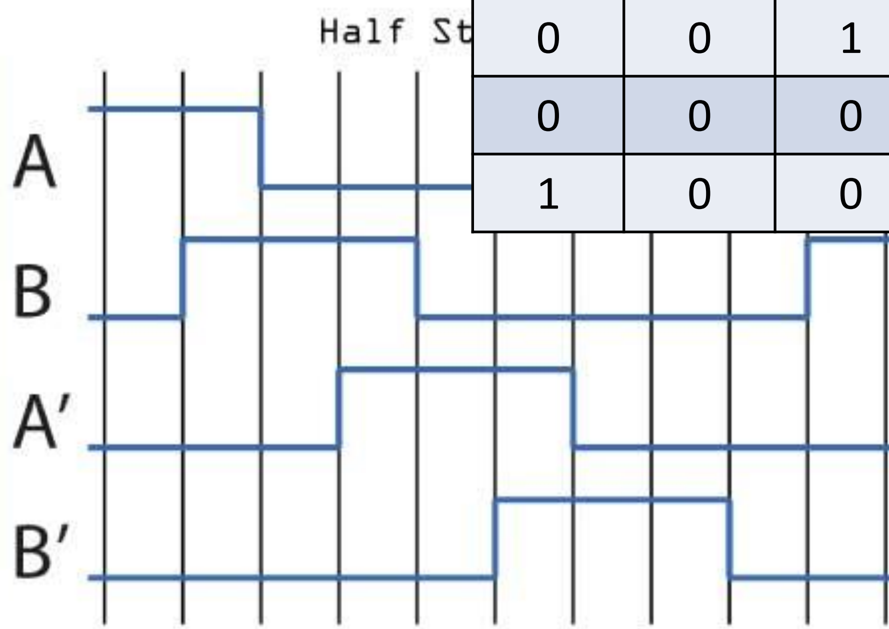


Half Step Drive

For increasing the resolution of the stepper we use the Half Step Drive mode. This mode is actually a combination of the previous two modes. Here we have one active coil followed by 2 active coils and then again one active coil followed by 2 active coils and so on. So with this mode we get double the resolution with the same construction. Now the rotor will make full cycle in 8 steps.

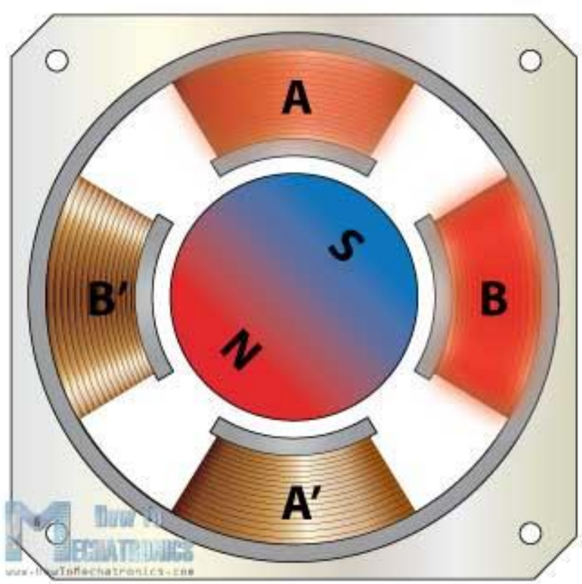
A	B	A'	B'
1	0	0	0
1	1	0	0
0	1	0	0
0	1	1	0
0	0	1	0
0	0	1	1
0	0	0	1
1	0	0	1

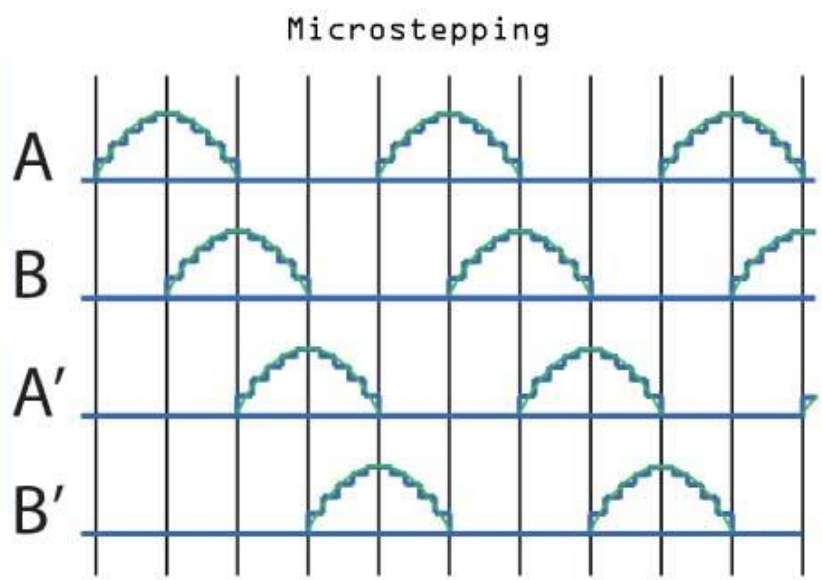




Microstepping

However the most common method of controlling stepper motors nowadays is the Microstepping. In this mode we provide variable controlled current to the coils in form of sin wave. This will provide smooth motion of the rotor, decrease the stress of the parts and increase the accuracy of the stepper motor.





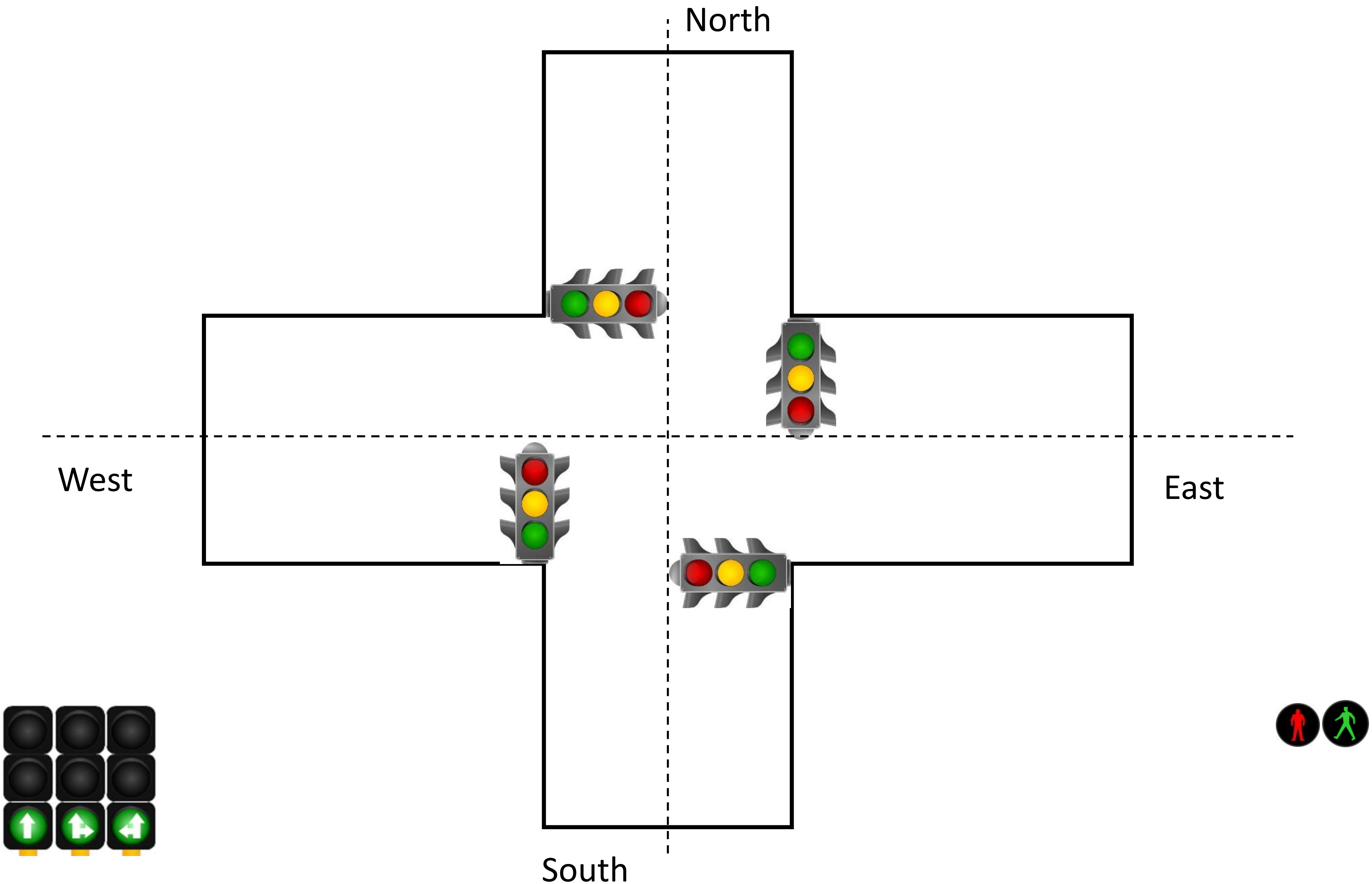
Step Angle

- Step angle = degree of motor rotation / pulse
  - 1.8^o 1 pulse
  - 360^o ? pulses
  - So $360^o / 1.8^o = 200$ pulses

  That means it requires 200 pulses to be applied to 1.8^o stepper motor to complete one revolution.
- Step angle = 360 / number of pulses

Traffic Control Interface

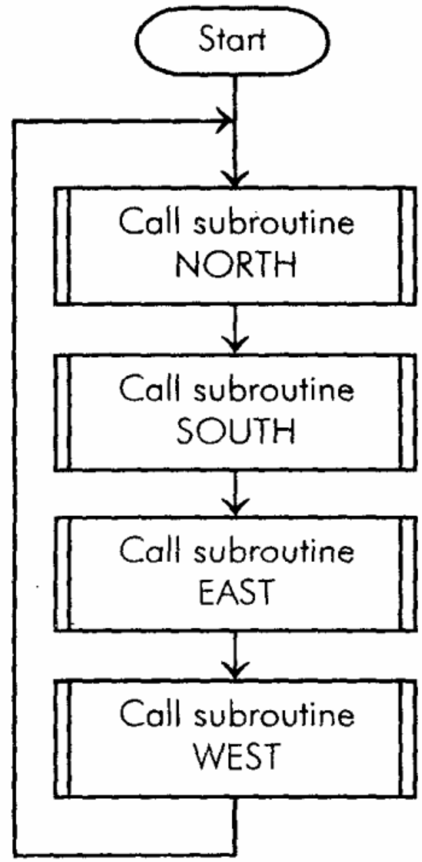
Traffic Control Interface



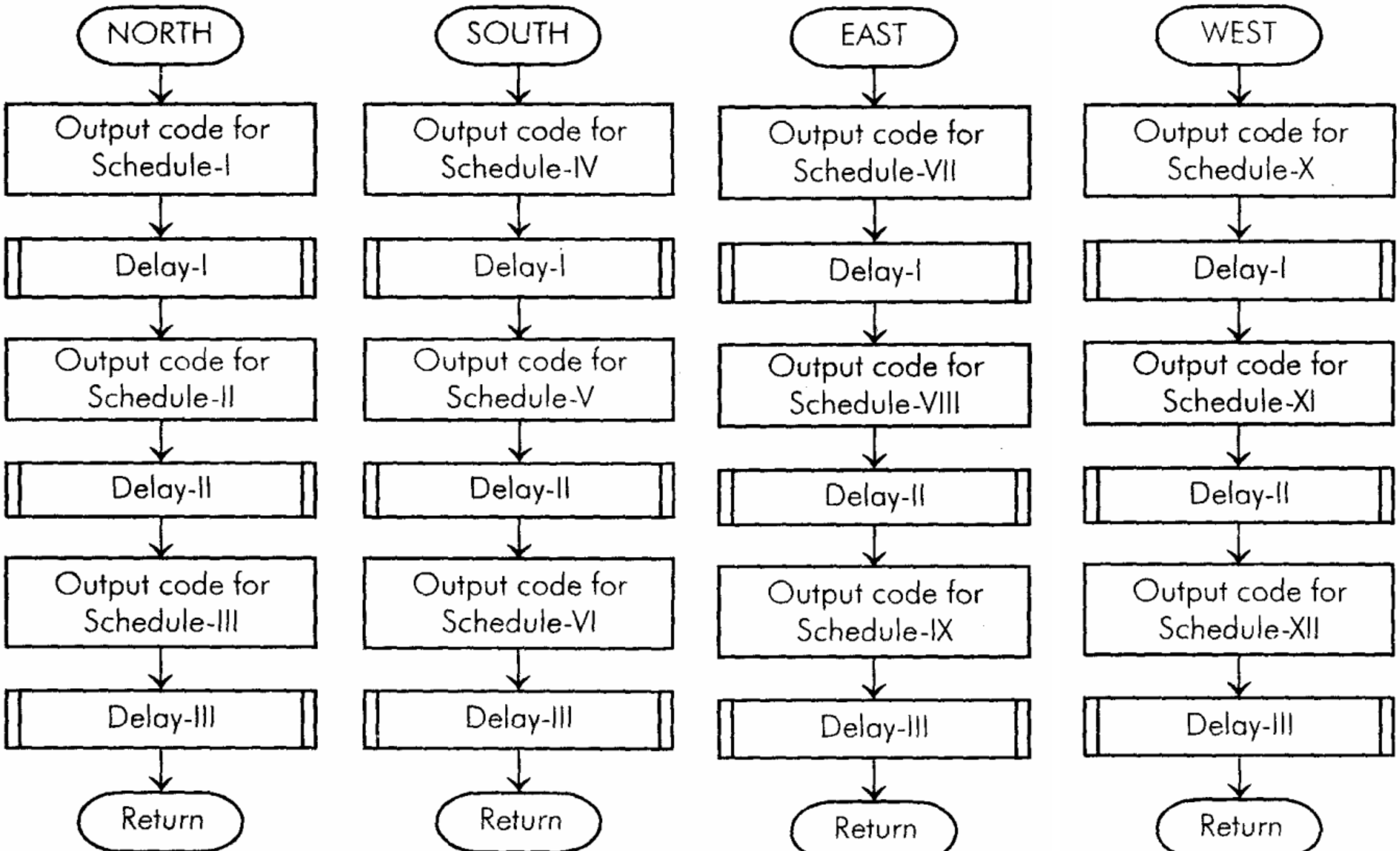
Switching Schedule for Traffic Lights

ON/OFF Status of Traffic Lights																				
SWITCHING SCHEDULE	PA_0 N_R	PA_1 N_Y	PA_2 N_G	PA_3 N_{FR}	PA_4 N_{FL}	PA_5 S_R	PA_6 S_Y	PA_7 S_G	PB_0 S_{FR}	PB_1 S_{FL}	PB_2 E_R	PB_3 E_Y	PB_4 E_G	PB_5 E_{FR}	PB_6 E_{FL}	PB_7 W_R	PC_0 W_Y	PC_1 W_G	PC_2 W_{FR}	PC_3 W_{FL}
SCHEDULE – I	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0
SCHEDULE - II	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
SCHEDULE - III	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
SCHEDULE - IV	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
SCHEDULE – V	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
SCHEDULE - VI	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0
SCHEDULE - VII	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
SCHEDULE - VIII	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0
SCHEDULE - IX	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
SCHEDULE - X	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0
SCHEDULE – XI	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0
SCHEDULE - XII	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1

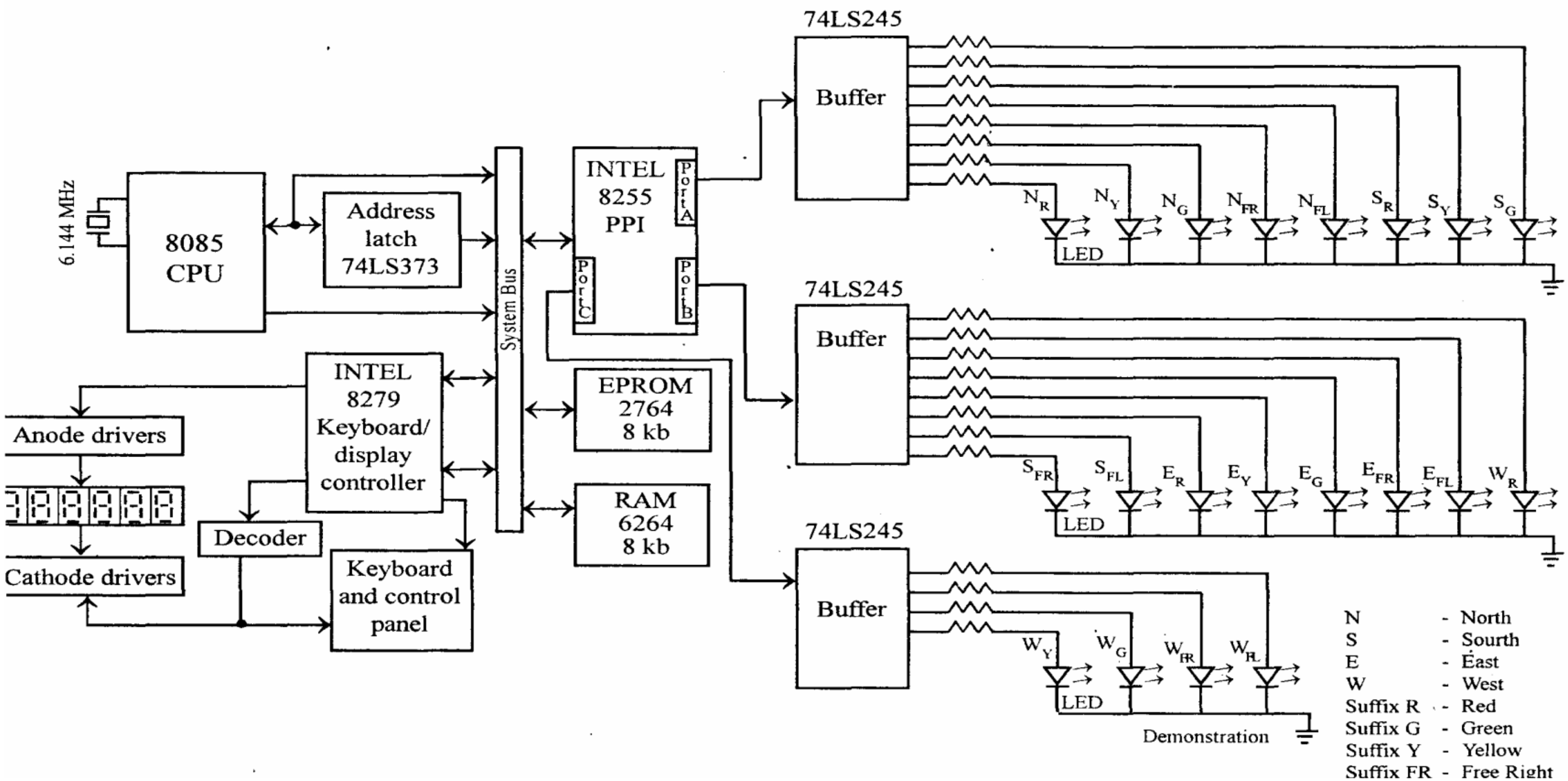
Flow Chart for Traffic Light Control Main Program



Flow Chart for Traffic Light Control Subroutine Program



Block Diagram – Traffic Light Control



8085 Microprocessor based Traffic Light Control demonstration System