Introduction to Relays

- A relay is a device that responds to a small current or voltage change by activating a switches or other devices.
- Used to remotely switch signals or power.
- Input control usually electrically isolated from output.
- Input signal determines whether switch is open or closed.

Various Relay Configurations

Types of Relays

- Classic general-purpose relay has an EM coil and can switch power.
- The solid-state relay (SSR) has an input-triggered semiconductor power switch.
- The read relay has an EM coil and can switch low level DC electronic signals.
- The bilateral switch uses CMOS, FET, or biFET transistors (technically not a relay but behaves similarly).
Electromagnetic Relay Basics

- Input circuit is an EM coil with an Iron Core.
- Output switch includes two sets of silver or silver-alloy contacts (called poles).
- One set is fixed to the relay frame, and other is located at end of leaf spring poles connected to the armature.
- Contacts held in “normally closed” position by the armature return spring.
- When input circuit energizes EM coil, a “pull-in” force is applied to the armature and “normally closed” contacts break while “normally open” contacts are made.

Solid State Relays

- Developed to solve limited life expectancy and contact bounce problems since they have no moving parts.
- Also, faster, insensitive to vibrations, reduced EMI, quieter, and no contact arcing.
- Optocoupler provides isolation between the input circuit (pseudocoil) and the triac (pseudocontact).
- Signal from phototransistor triggers the output triac so that it switches the load current.
- Zero-voltage detector triggers triac only when AC voltage is zero, reducing surge currents when triac is switched.
- Once triggered, triac conducts until next zero crossing.
Pulse-Width Modulated DC Motors

- DC motor also has frame that remains motionless and an armature that moves in this case in a circular manner.
- When current flows through EM coil, magnetic force created that causes rotation of the shaft.
- Brushes positioned between frame and armature used to alternate the current direction through the coil so that a DC current generates a continuous rotation of the shaft.
- When current removed, shaft is free to rotate.
- Pulse-width modulated DC motor activated with fixed magnitude current but duty cycle varied to control speed.

Interfacing EM Relays, Solenoids, and DC Motors

- Interface circuit must provide sufficient current and voltage to activate the device.
- In off state, input current should be zero.
- Due to inductive nature of the coil, huge back electromotive force (EMF) when coil current is turned off.
- Due to high speed transistor switch, there is a large \( \frac{di}{dt} \) when the coil is deactivated (activation also but smaller).
- Voltages can range from 50 to 200V.
- To protect the driver electronics, a snubber diode is added to suppress the back EMF.
Relay and Motor Interfaces

Isolated Interfaces

H-Bridge

Isolated H-Bridge with Direction Control
Stepper Motors

- Very popular due to inherent digital interface.
- Easy to control both position and velocity in an open-loop fashion.
- Though more expensive than ordinary DC motors, system cost is reduced as they require no feedback sensors.
- Used in disk drives and printers.
- Can also be used as shaft encoders to measure both position and speed.

Stepper Motor Sequence

<table>
<thead>
<tr>
<th>PortB</th>
<th>A</th>
<th>A’</th>
<th>B</th>
<th>B’</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Activate</td>
<td>deactivate</td>
<td>activate</td>
<td>deactivate</td>
</tr>
<tr>
<td>9</td>
<td>Activate</td>
<td>deactivate</td>
<td>deactivate</td>
<td>activate</td>
</tr>
<tr>
<td>5</td>
<td>Deactivate</td>
<td>activate</td>
<td>deactivate</td>
<td>activate</td>
</tr>
<tr>
<td>6</td>
<td>Deactivate</td>
<td>activate</td>
<td>activate</td>
<td>deactivate</td>
</tr>
</tbody>
</table>

Simple Stepper Motor Interface

Stepper Motor Basic Operation
Slide 25

**Stepper Motor Basic Operation (cont)**

Slide 26

**Unipolar Stepper Motor Interface**

Slide 27

**Slip**
- A *slip* is when computer issues a sequence change, but the motor does not move.
- Occurs if load on shaft exceeds available torque of motor.
- Can also occur if computer changes output too fast.
- If initial shaft angle known and motor never slips, computer can control shaft angle and speed without position sensor.

Slide 28

**Linked List Data Structure**
Linked List to Control Stepper Motor

;Linked list stored in EEPROM
S10 fcb 10 ;Output pattern
    fdb S9 ;Next if CW
    fdb S6 ;Next if CCW
S9 fcb 9
    fdb S5
    fdb S10
S5 fcb 5
    fdb S6
    fdb S9
S6 fcb 6
    fdb S10
    fdb S5
;Global variables stored in RAM
POS rmb 1 ;0<=POS<=199
PT rmb 2 ;to current state

Helper Functions to Control Stepper Motor

;Move 1.8 degrees clockwise
CW: ldx PT ;current state
    ldx 1,X ;next clockwise
    stx PT ;update pointer
    ldaa ,X ;output pattern
    staa PORTB ;set phase control
    ldaa POS ;update position
    inca ;clockwise
    cmpa #200
    bne OK1 ;0<=POS<=199
    clra
OK1: staa POS
    rts

Helper Functions to Control Stepper Motor

;Move 1.8 degrees counterclockwise
CCW: ldx PT ;current state
    ldx 3,X ;next CCW
    stx PT ;update pointer
    ldaa ,X ;output pattern
    staa PORTB ;set phase control
    ldaa POS ;update position
    deca ;CCW direction
    cmpa #255
    bne OK2 ;0<=POS<=199
    ldaa #199
OK2: staa POS
    rts
Helper Functions to Control Stepper Motor

const struct State
{
    unsigned char Out; /* Output for this state */
    const struct State *Next[2]; /* Next state */
} StatePtr;

unsigned char POS; /* b/w 0 and 199, shaft angle */

#define clockwise 0 /* Next index*/
#define counterclockwise 1 /* Next index*/

StateType fsm[4]= { {10,{&fsm[1],&fsm[3]}},
    { 9,{&fsm[2],&fsm[0]}},
    { 5,{&fsm[3],&fsm[1]}},
    { 6,{&fsm[0],&fsm[2]} } };  
StatePtr Pt; /* Current State */

void CW(void){
    Pt=Pt->Next[clockwise];
    PORTB=Pt->Out;
    if(POS++==200) POS=0;}

void CCW(void){
    Pt=Pt->Next[counterclockwise];
    PORTB=Pt->Out;
    if(POS==0) POS=199;
    else POS--; }

void Init(void){
    POS=0;
    Pt=&fsm[0];}
High-Level Control of Stepper Motor

```c
void SEEK(unsigned char New){
    int CWsteps,i;
    if((CWsteps=New-POS)<0)
        CWsteps+=200;
    if(CWsteps>100)
        for(i=CWsteps;i<200;i++)
            CCW();
    else
        for(i=0;i<CWsteps;i++)
            CW(); }
```

Stepper Motor as Shaft Position Sensor

Timing of Stepper Motor as Shaft Position Sensor