```
/* SELFTEST FOR PIC18F4550 Automated-Guided Vehicle CONTROLLER
   Converted on 16 Aug 2009 VER02 from SELFTEST.ASM
   The SELFTEST Mode depends on the button pressed during RESET
   i.e. Press Button first then RESET
```

1. (no switches pressed) = LED test
2. Switch RB0 = Stepper and DC Motor Test note:
   After this, RB1 and RB2 move the DC motor
3. Switch RB1 = LED test

```
/* Program starts here: */
#include <p18f4550.h>
#include <delays.h>

#define     RB0     PORTBbits.RB0       /* labelled on PIC board */
#define     RB1     PORTBbits.RB1
#define     RB2     PORTBbits.RB2

#define SMEnable    PORTEbits.RE0       /* Stepper motor enable bit */
#define SMReset     PORTEbits.RE1       /* Stepper motor reset bit */

#define DCMEnable   PORTEbits.RE2       /* DC motor enable bit */
#define DCMPhase    PORTCbits.RC7       /* DC motor reset bit */

/* Left-Motor */
#define SMLDir      PORTCbits.RC2       /* Left stepper motor direction */
#define SMLClk      PORTCbits.RC0       /* Left stepper motor clock */

/* Right-Motor */
#define SMRDir      PORTCbits.RC6       /* Right stepper motor direction */
#define SMRClk      PORTCbits.RC1       /* Right stepper motor clock */
```

Defines all PIC-18 labels

Defines all delay functions

Defines the symbol for the string on the right.

(Whenever the program encounters the symbol thereafter, it will be replaced by the string.)

```
#define ONb    0                    /* active low */
#define ON     1                    /* active high */
#define OFFb   1                    /* active low */
#define OFF    0                    /* active high */
#define TRUE   1
#define TransistorON     PORTBbits.RB5
#define SensorClk        PORTBbits.RB3
```

Active HIGH: 1=on, 0=off

Active LOW: 0=on, 1=off

```
void            LEDTest();
void            sensortest();
void            motortest();
```

Function Prototypes

Needed if the function is used (e.g. in main) before being defined.

```
void                main()
{
      /*  ===== Use with SP PIC18f4550 board with USB bootloader ==== */

    ADCON1 = 0x0F;          /* PORTA all digital, use Vcc & Gnd for A/D ref */
    CMCON = 0x07;           /* disable analogue comparator */
      /*  end PIC18F4550  */

      /* Initialise PORTs */

    TRISC = 0xFF;
    TRISD = 0x00;
    TRISB = 0x07;

    SMEnable = OFF;
    DCMEnable = OFF;                    /* Disable DC Motor */

    if (RB0 == ONb)                     /* RB0 pressed, do Motor test */
        motortest();


    if (RB1 == ONb)                     /* RB1 pressed, do sensor test */
        sensortest();

    /* otherwise this is the LED test */
    LEDTest();

}
```

PORTC – all inputs

PORTD – all outputs (to LEDs)

PORTB – bit 0-2: inputs, other bits: outputs

```
void                LEDTest()
{
    unsigned int    i;
    TRISD = 0x00;                   /* all outputs */

    while (TRUE) {
        PORTD = 0 b11110000;        /* leftmost LED - bit 4 on */

        for (i = 0; i < 6; i++) {

            Delay10KTCYx(250);
            Delay10KTCYx(250);
            PORTD = PORTD << 1; /* shift bits */
        }
    }
}
```

Repeat forever:

PORTD – bit 3-7 drive the 5 LEDs on the Interface Board, with bit 3 on initially.

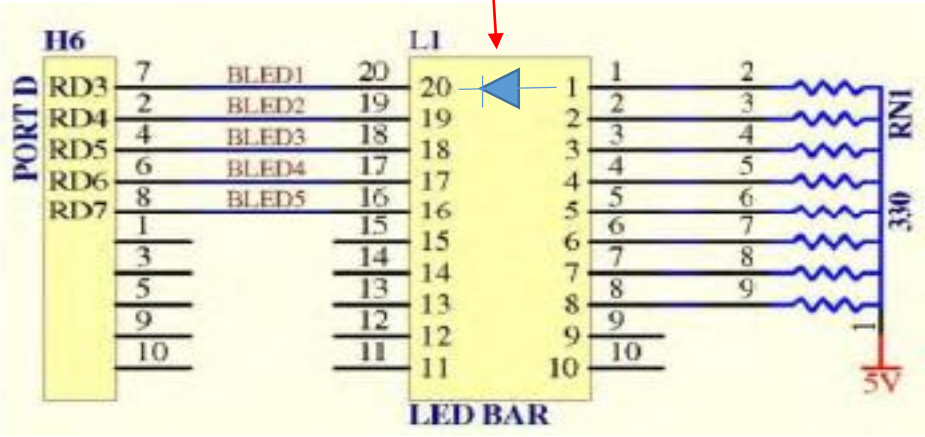
('0' will turn on the LED.)

From 0 to 5: i.e. 6 times

Delay for 0.4s

Shift PORTD pattern left by 1 bit.

Alternatively: ***PORTD <<= 1;***



```
void            sensortest()
{
    unsigned char   LEDData;     /* hold LED data */
    TRISB = 0x00;
    TRISD = 0x00;

    while (TRUE) {

        TransistorON = ON;       /* TURN ON EMITTER */
        SensorClk = 0;
        SensorClk = 1;

        LEDData = PORTA & 0x1f; /* READ IN */

        LEDData = ~LEDData;      /* INVERT DATA */
        LEDData = LEDData << 3; /* using RD3-RD7 to display */
        PORTD = LEDData;         /* DISPLAY */
        // STATUS ON LED
        Delay10KTCYx(100);       /* DELAY */

    }
}
```

Repeat forever:

PORTB – all outputs

PORTD – all outputs (to LEDs)

Enable the sensors by turning on the IR LEDs.

Clock pulse to latch the data for PORTA.

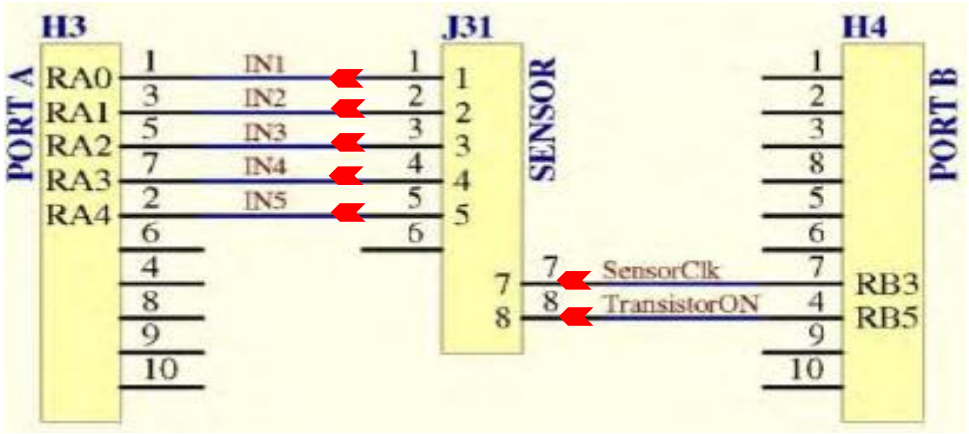
00011111 – Capture only the lower 5 bits, RA0-RA5.

Bitwise AND

Alternatively:

PORTD = ~(PORTA<<3) & 0b11111000;

Shift left by 3 bit.



```
/* ROUTINE TO RUN THE TWO STEPPER MOTOR AND DC MOTOR */
void            motortest()
{
    unsigned char   savx;           /* for counting */

/* Enable and reset L297 */

    TRISB = 0x07;
    TRISC = 0x00;
    TRISE = 0;
    SMReset = 1;
    SMEnable = 1;
    SMReset = 0;                    // RESET PULSE

    SMReset = 1;                    /* Reset Stepper Motor driver */

    SMLDir = 0;                     /* SET DIR-X TO CCW */
    SMLClk = 0;                     /* SET CLK-X TO LO */

    SMRDir = 0;                     /* SET DIR-Y TO CCW */
    SMRClk = 0;                     /* SET CLK-Y TO LO */

    PORTD = 0;                      /* CLEAR LED */
    savx = 0;                       /* clear counter */
```

PORTB – bit 0-2: inputs, others: outputs

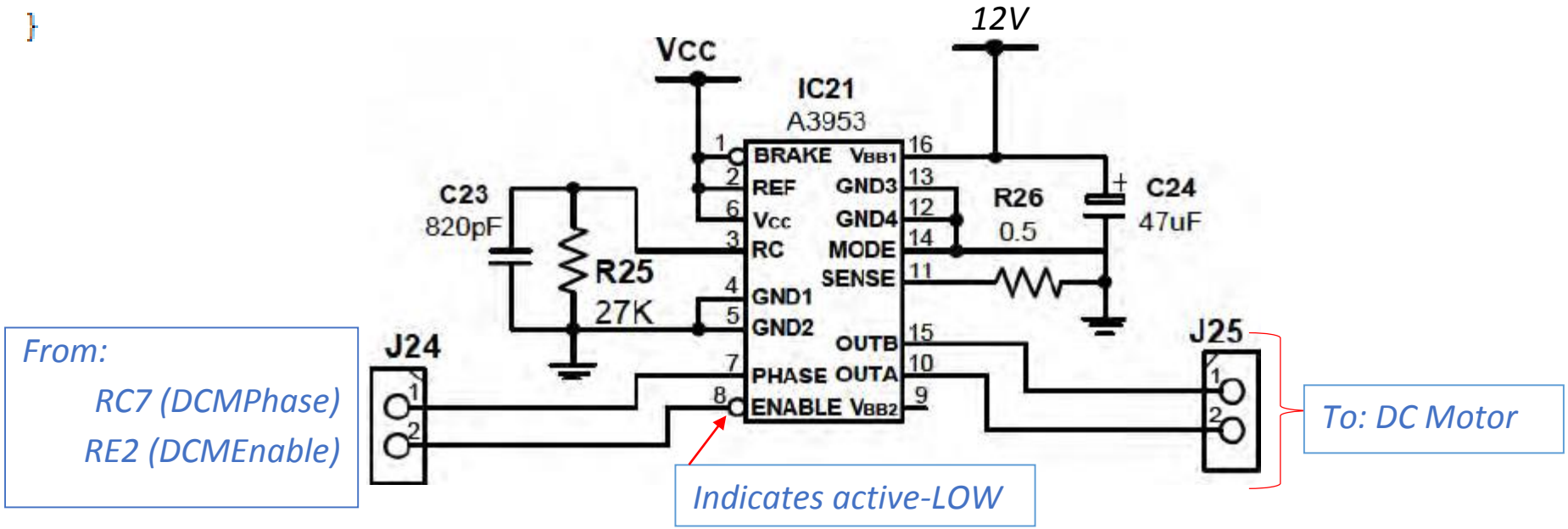
PORTC – all outputs

PORTE – all outputs

```
/* Start to run motors */
while (TRUE) {

    /* check switches for DC motor operation first */
    if ((SW2 == ONb) && (SW3 == OFFb)) {     /* RB1 on, RB2 off */
        DCMPhase = ON;              /* CW */
        DCMEnable = ONb;
    }
    if ((SW2 == OFFb) && (SW3 == ONb)) {     /* RB1 off, RB2 on */
        DCMPhase = OFF;             /* CCW */
        DCMEnable = ONb;
    }
    if ((SW2 == OFFb) && (SW3 == OFFb)) {    /* RB1 off, RB2 off */
        DCMPhase = OFF;         /* disable motor */
        DCMEnable = OFFb;
    }
```

Repeat forever:

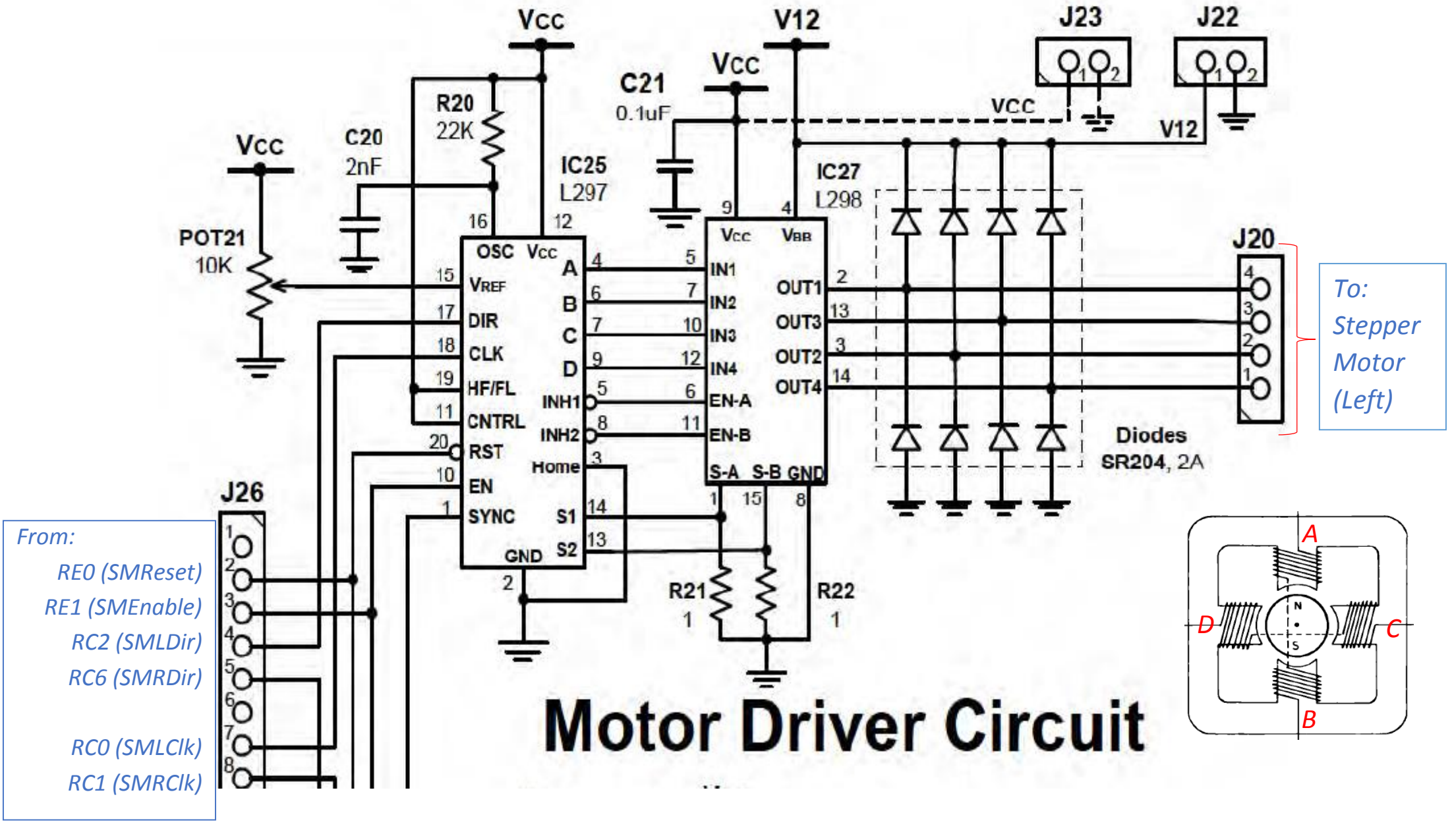


```
        /* Check for stepper motor */
        /* Move left motor 1 step */
        SMLDir = 0;                 /* CCW */
        SMLClk = 1;
        Delay10KTCYx(1);            /* 0.833 ms */
        SMLClk = 0;
        Delay10KTCYx(1);            /* 0.833 ms */

        /* Move right motor 1 step */
        SMRDir = 0;                 /* CCW */
        SMRClk = 1;
        Delay10KTCYx(1);
        SMRClk = 0;
        Delay10KTCYx(1);

        /* Displays a scrolling pattern on the LEDBAR */
        savx++;             /* increment */
        PORTD = savx;       /* and show */
    }
}
```

Each clock pulse will change pattern ABCD of the stepper motor and rotate it by 1 step.


Vcc
V12
J23
J22
C21
0.1uF
R20
22K
C20
2nF
IC25
L297
IC27
L298
POT21
10K
OSC
Vcc
VREF
DIR
CLK
HF/FL
CNTRL
RST
EN
SYNC
GND
A
B
C
D
INH1
INH2
Home
S1
S2
IN1
IN2
IN3
IN4
EN-A
EN-B
S-A
S-B
GND
VBB
OUT1
OUT3
OUT2
OUT4
J20
J26
R21
R22
Diodes
SR204, 2A
To:
Stepper
Motor
(Left)
From:
RE0 (SMReset)
RE1 (SMEnable)
RC2 (SMLDir)
RC6 (SMRDir)
RC0 (SMLClk)
RC1 (SMRClk)
A
B
C
D
N
S
Motor Driver Circuit