

'-----Title-----

' File.....compass_car.pbp
' Started....4/26/07
' Microcontroller used: Microchip Technology 16F84A
' microchip.com
' PBPro Code, micro-Engineering Labs, Inc.
' melabs.com

'-----Program Description-----

' This program takes the inputs from the Dinsmore compass device, and
' uses that info to power servos. This turns the compass to face magnetic
' north, and it will go straight. If you turn it away from north, it will
' turn back to it, and keep going straight.

'-----Constants/Defines-----

'-----Variables-----

c0 VAR BYTE

buzzer VAR PORTC.5

buzzer1 VAR PORTC.6

northT VAR PORTC.7 'Sets the variable c0, so that the
program can recognize it, and use it.

'-----Initialization-----

TRISB = %10010110 'Sets up pins B0-B3 of PORTB as an input

TRISC = %10000000 'and pins B4-B7 as outputs

PORTC = %00000000

PORTB = %00000000 'Sets all PORTB pins to an active low.

PORTC = %00000000

INTCON = %00000001

'-----Main Code-----

PAUSE 1000

ON INTERRUPT GOTO beep

loop:

IF PORTB.1 = 0 AND PORTB.2 = 0 AND PORTB.4 = 1 AND northT = 1 THEN
southwest

IF PORTB.1 = 0 **AND** northT = 0 **AND** PORTB.2 = 1 **AND** PORTB.4 = 1 **THEN**
northwest

IF PORTB.4 = 0 **AND** PORTB.2 = 0 **AND** PORTB.1 = 1 **AND** northT = 1 **THEN**
southeast

IF PORTB.4 = 0 **AND** northT = 0 **AND** PORTB.2 = 1 **AND** PORTB.1 = 1 **THEN**
northeast

IF PORTB.1 = 0 **AND** PORTB.2 = 1 **AND** PORTB.4 = 1 **AND** northT = 1 **THEN**
west

*'If the west LED is on, the condition is true, and the program goes to
'the label "left".*

IF PORTB.2 = 0 **AND** PORTB.1 = 1 **AND** PORTB.4 = 1 **AND** northT = 1 **THEN**
south

*'If the south LED is on, the condition is true, and the program goes
to
'the label "right".*

IF PORTB.4 = 0 **AND** PORTB.1 = 1 **AND** PORTB.2 = 1 **AND** northT = 1 **THEN**
east

*'If the east LED is on, the condition is true, and the program goes to
'the label "right".*

IF northT = 0 **AND** PORTB.1 = 1 **AND** PORTB.2 = 1 **AND** PORTB.4 = 1 **THEN**
north

*'If the north LED is on, the condition is true, and the program goes
to
'the label "north". It will only go north if the preceding three
'conditions are false.*

DISABLE

south:

FOR c0 = 1 **TO** 10 *'FOR-NEXT loop is to be run 10 times.*

LCDOUT \$FE,1,"South" *'Clears LCD screen, displays South*

PULSOUT 6, 100

PULSOUT 7, 100

PAUSE 20 *'20 millisecond pause*

NEXT *'Sends it to the next cycle of the FOR-NEXT loop*

GOTO loop *'Sends it to the label "loop"*

southwest:

```
FOR c0 = 1 TO 10  'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"South West"  'Clears LCD screen, displays West
PULSOUT 6, 100
PULSOUT 7, 100
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

west:

```
FOR c0 = 1 TO 10  'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"West"  'Clears LCD screen, displays West
PULSOUT 6, 100
PULSOUT 7, 100
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

northwest:

```
FOR c0 = 1 TO 10  'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"North West"  'Clears LCD screen, displays North West
PULSOUT 6, 100
PULSOUT 7, 100
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

north:

```
FOR c0 = 1 TO 10  'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"North"  'Clears LCD screen, displays North
PULSOUT 6, 100
```

```
PULSOUT 7, 200
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

northeast:

```
FOR c0 = 1 TO 10 'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"North East" 'Clears LCD screen, displays North East
PULSOUT 6, 200
PULSOUT 7, 200
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

east:

```
FOR c0 = 1 TO 10 'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"East" 'Clears LCD screen, displays East
PULSOUT 6, 200
PULSOUT 7, 200
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

southeast:

```
FOR c0 = 1 TO 10 'FOR-NEXT loop is to be run 10 times.
LCDOUT $FE,1,"South East" 'Clears LCD screen, displays South East
PULSOUT 6, 200
PULSOUT 7, 200
PAUSE 20          '20 millisecond pause
NEXT              'Sends it to the next cycle of the FOR-NEXT loop
GOTO loop        'Sends it to the label "loop"
```

beep:

```
FOR c0 = 1 TO 100
  IF c0 < 10 THEN buzz1
  IF 10 < c0 < 20 THEN buzz2
  IF 20 < c0 < 30 THEN buzz1
  IF 30 < c0 < 40 THEN buzz2
  IF 40 < c0 < 50 THEN buzz1
  IF 50 < c0 < 60 THEN buzz2
  IF 60 < c0 < 70 THEN buzz1
  IF 70 < c0 < 80 THEN buzz2
  IF 80 < c0 < 90 THEN buzz1
  IF 90 < c0 < 100 THEN buzz2
```

buzz1:

```
LOW buzzer1
```

```
HIGH buzzer
```

buzz2:

```
LOW buzzer
```

```
HIGH buzzer1
```

```
LCDOUT $FE,1, "Backwards"
```

```
PULSOUT 6,200
```

```
PULSOUT 7,100
```

```
NEXT
```

```
GOTO avoid
```

avoid:

```
PULSOUT 6,100
```

```
PULSOUT 7,100
```

```
NEXT
```

GOTO loop

RESUME *'Go back to next command in the
'main program before the
'interrupt was triggered.*

ENABLE *'Enables interrupts after handler*

END *'Ends the program*