SHADOW FAN

Lalit Damodaran
Mrudula Vaidya
Pratyush Bangera

CONTENTS

INTRODUCTION

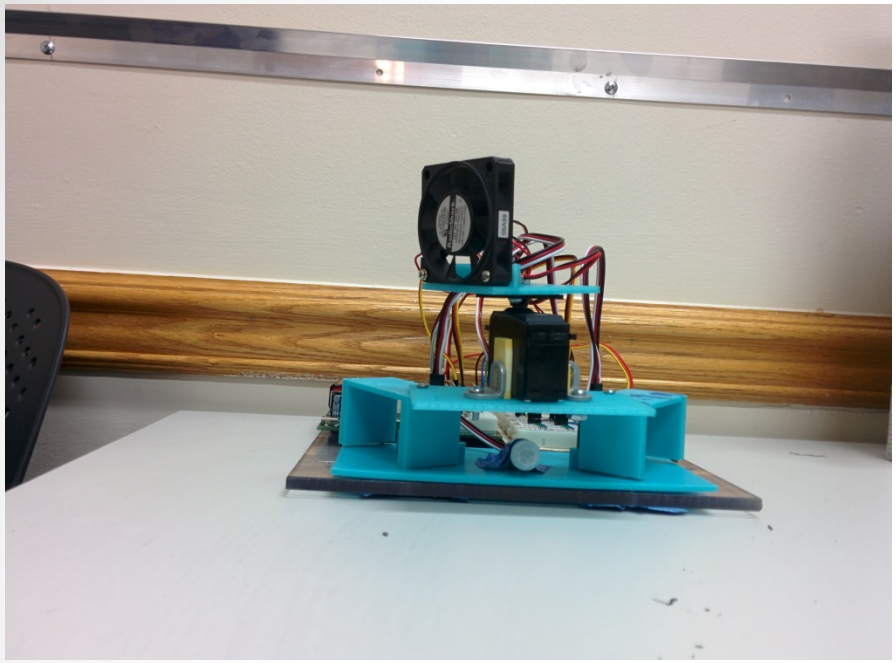
- Standard fans are widely used in personal office spaces.
- Typically provided with 2 modes – Static and a Preset swing mode
- The first mode requires physical effort for the fan to be directed to the changing position of the user
- The second mode can ne wasteful as the fan may be directed to empty spaces due to preset swing.

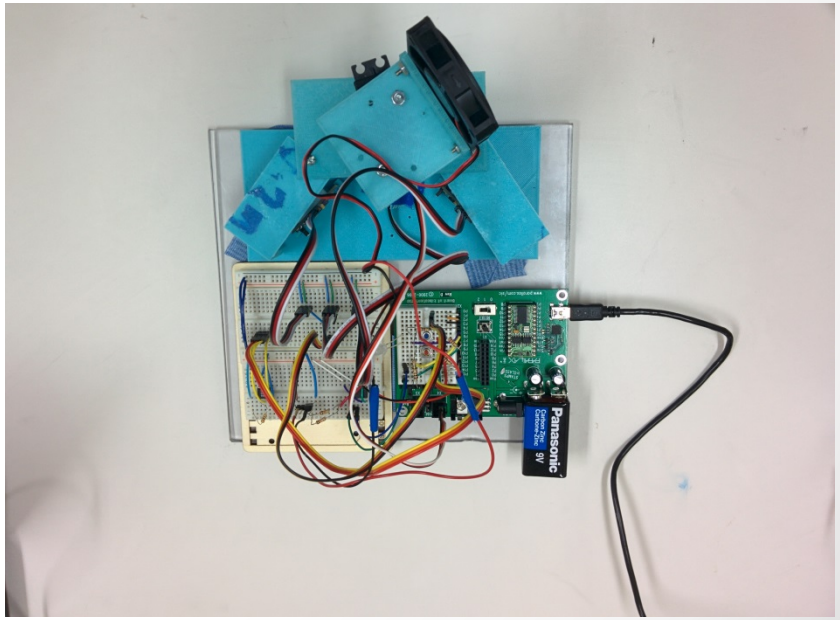
GOALS

- Design an automated fan that follows the user's movement.
- Change the speed of the fan depending on the ambient temperature
- In case of more than a single user, provide automated oscillation of the pan, based on their positions.

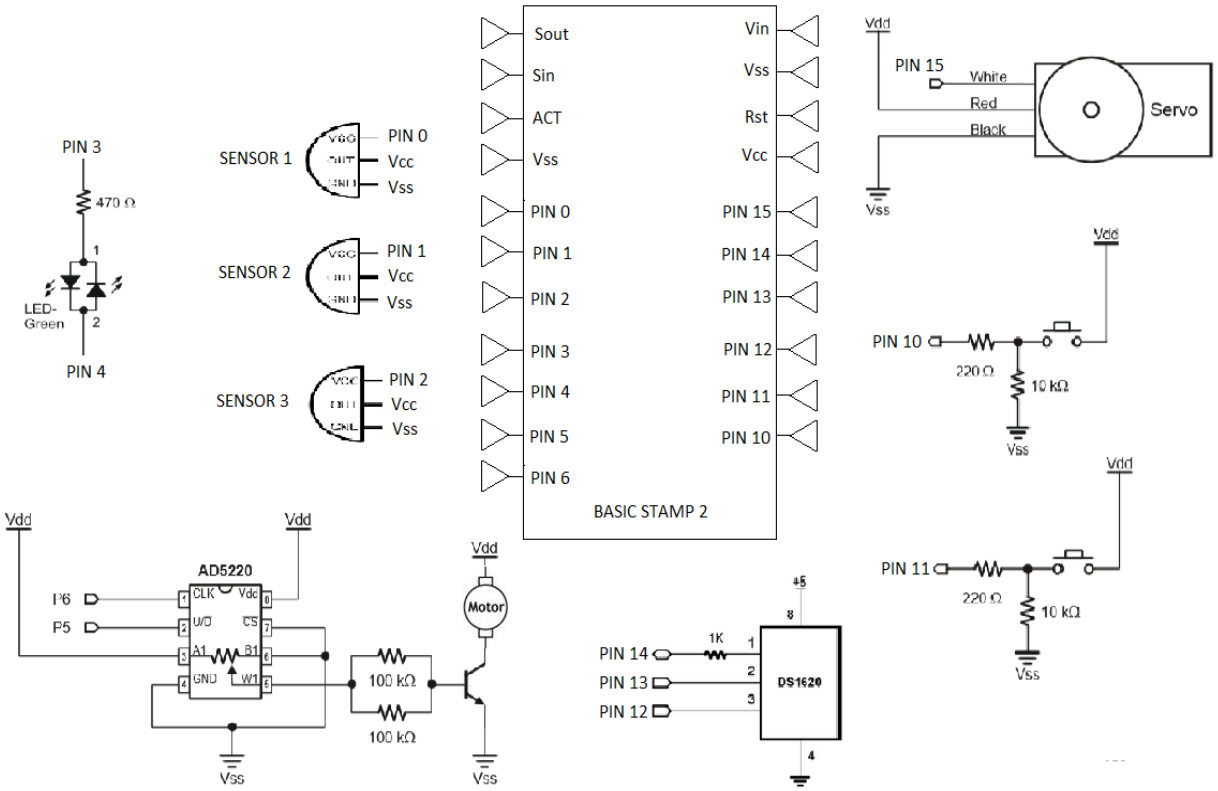
CONSTRUCTION

- Components used:
  - BASIC STAMP 2 Microcontroller
  - Parallax Standard Servomotor
  - PIR (Passive Infrared) Sensors
  - Digital Potentiometer (AD5220)
  - Transistor
  - Temperature Sensor (DS1620)




Panasonic
6V

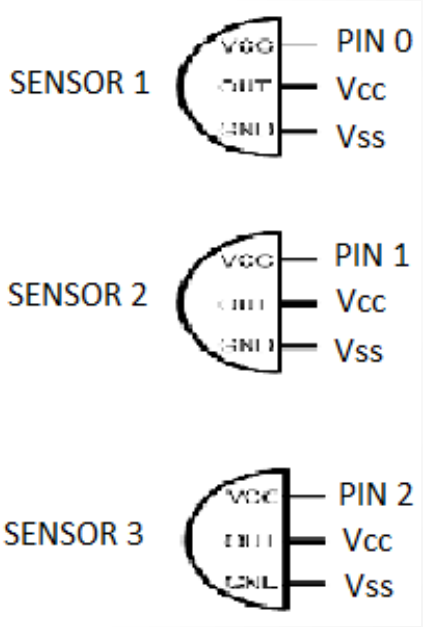
CIRCUIT LAYOUT



CIRCUITS

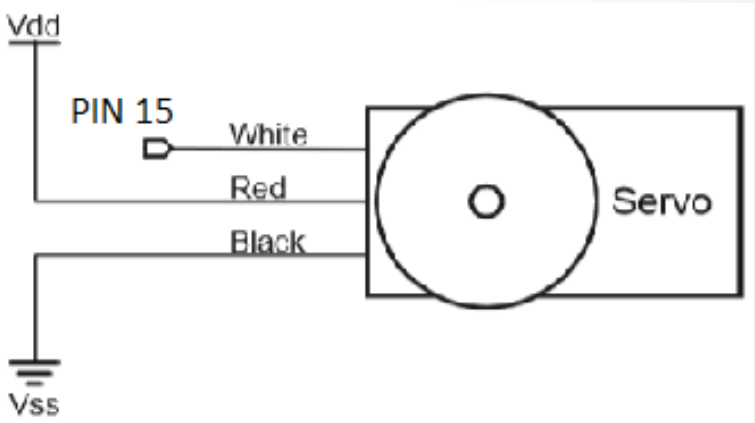
- **PIR (Passive Infrared) Sensor**

- The PIR sensor will give a HIGH or LOW signal depending on whether it can detect motion or not.



- **Standard Servomotor**

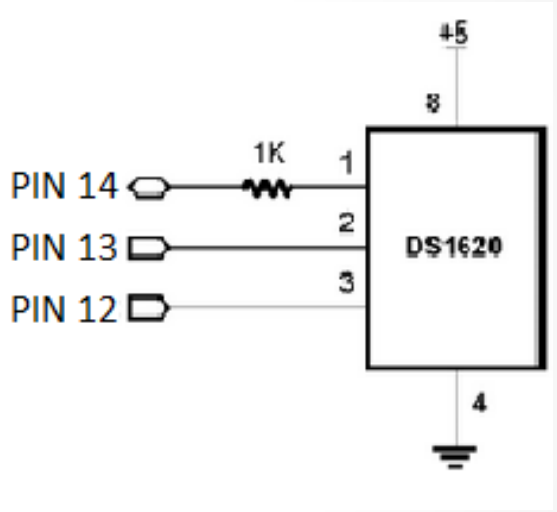
- Based on the sensor that detects the motion, the servo head, carrying the fan, is moved to that particular position



- **Temperature Sensor (DS1620)**

- The output of the temperature sensor will indicate the ambient temperature.

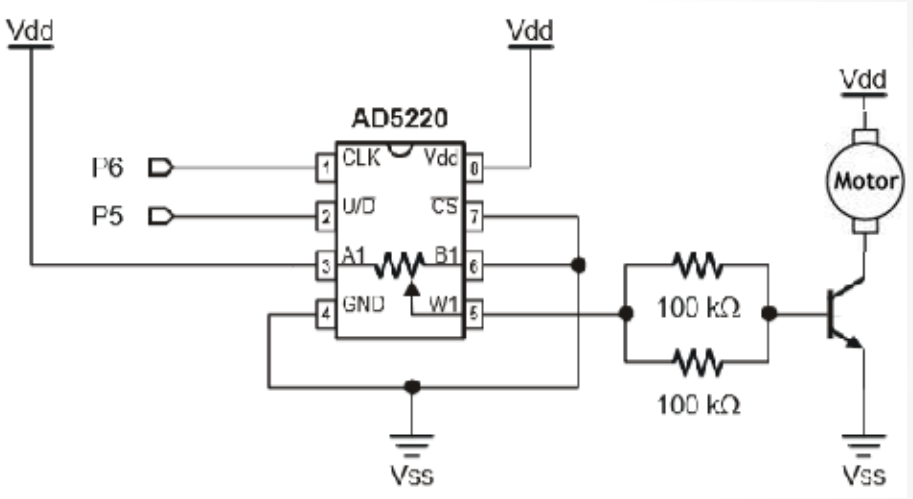
- The speed of the fan is varied accordingly with respect to the increase or decrease in the temperature.



- **Digital Potentiometer (AD5220)**

  - The potentiometer is used to vary the speed of the fan by varying the resistance

  - With every degree increase, the potentiometer is varies by 20 steps.



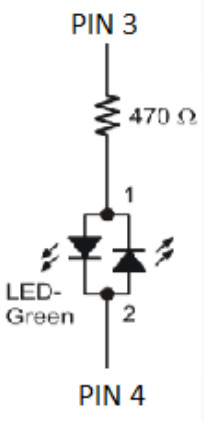
- Miscellaneous Circuits
  - **Bicolor LED**

    Green – Increasing Temperature

    Red – Decreasing Temperature

  - **Brushless DC motor (Fan)**

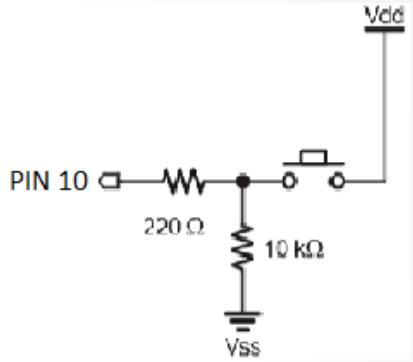
    The output of the digital potentiometer is used to run the fan and vary the speed depending on the change in temperature



- Pushbuttons

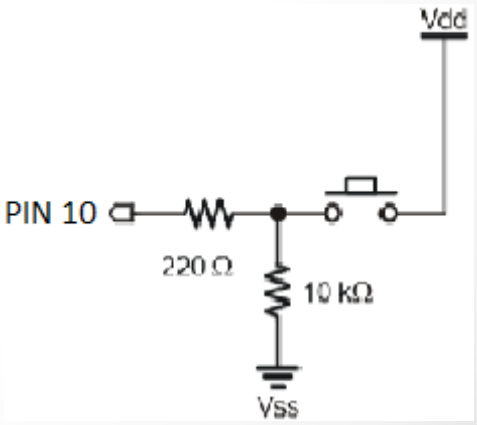
  Start Switch -

  Used to switch the circuit on or to restart the circuit.

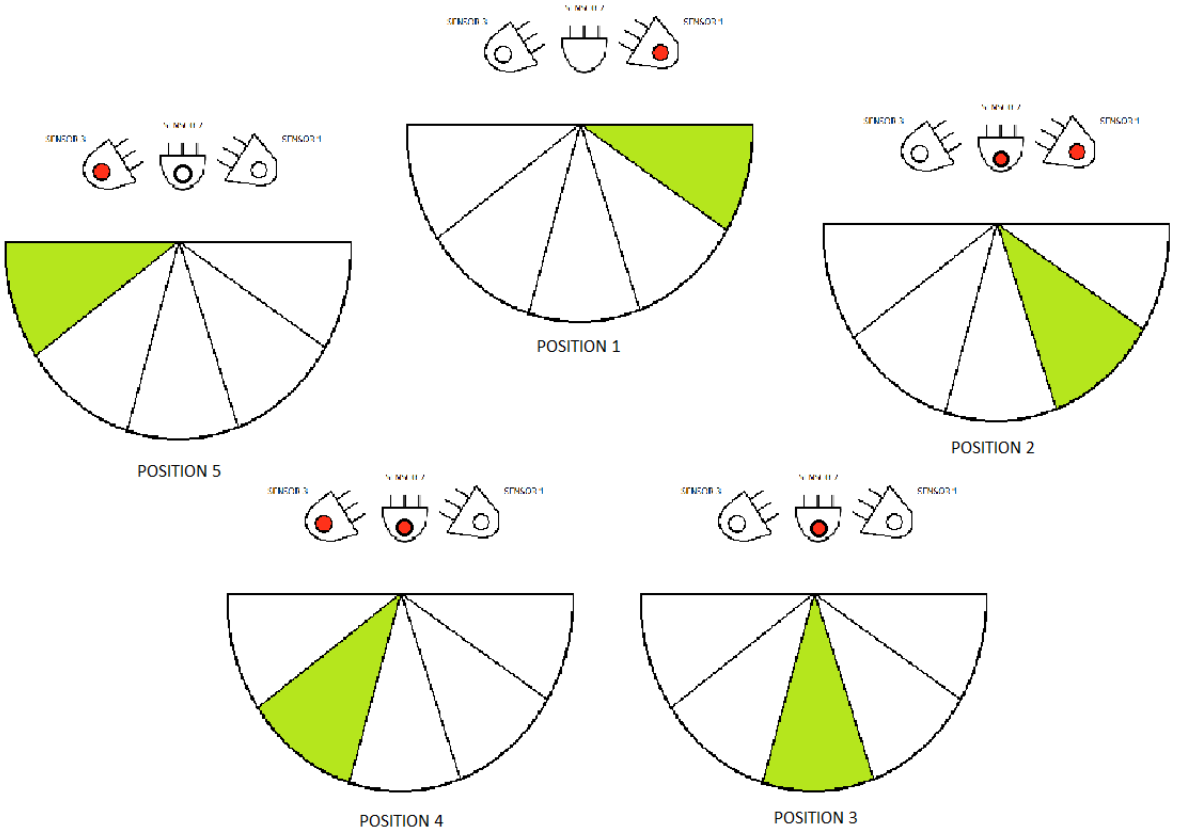


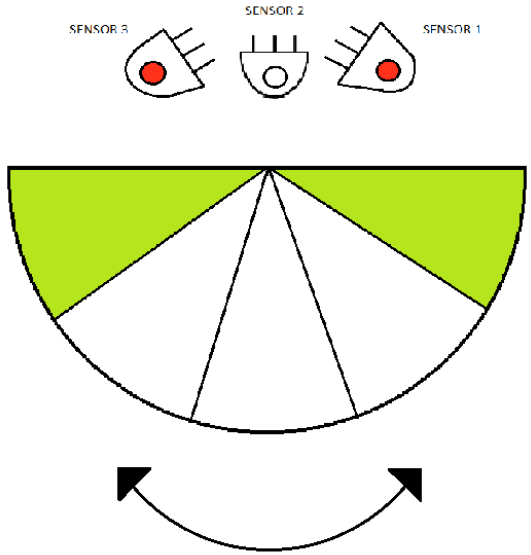
  Kill Switch-

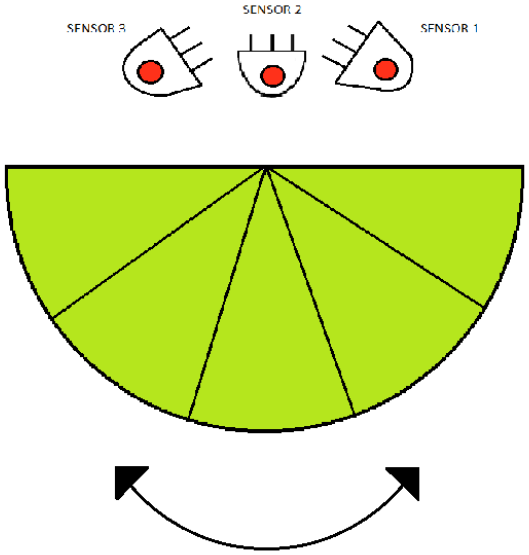
  Used to stops the motion of the servo.



WORKING

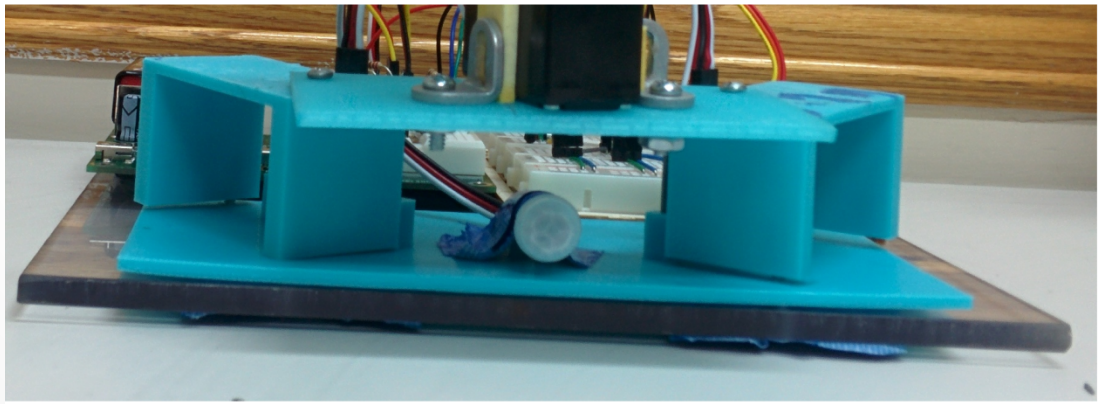



SENSOR 3
SENSOR 2
SENSOR 1


SENSOR 3
SENSOR 2
SENSOR 1

PROBLEMS FACED

- **Sensor Calibration**
  - Difficulty in positioning the PIR sensors on the board to eliminate superposition of their sensing range
  - Overcome by designing a mechanical structure separating the three sensors



- **Temperature sensitivity output**
  - No observable change in fan speed corresponding to varying temperature
  - Solved by using an LED which shows the incorporation of temperature change on the fan speed

BASIC STAMP CODE

```
' {$STAMP BS2}
' {$PBASIC 2.5}



DQ              CON     14                      ' DS1620.1 (data I/O)
Clock           CON     13                      ' DS1620.2
Reset           CON     12                      ' DS1620.3
' -----------------------------------------------------------------------------
' Constants ' ----------------------------------------------------------------------------
RdTmp           CON     $AA                     ' read temperature
WrHi            CON     $01                     ' write TH (high temp)
WrLo            CON     $02                     ' write TL (low temp)
RdHi            CON     $A1                     ' read TH
RdLo            CON     $A2                     ' read TL
StartC          CON     $EE                     ' start conversion
StopC           CON     $22                     ' stop conversion
WrCfg           CON     $0C                     ' write config register
RdCfg           CON     $AC                     ' read config register
' -----------------------------------------------------------------------------
' Variables ' ----------------------------------------------------------------------------
tempIn          VAR     Word                    ' raw temperature
sign            VAR     tempIn.BIT8             ' 1 = negative temperature
tSign           VAR     Bit
tempC           VAR     Word                    ' Celsius
tempC1          VAR     Word
tempF           VAR     Word                    ' Fahrenheit
counter         VAR     Byte
counter1        VAR     Byte
counter2        VAR     Byte
counter3        VAR     Byte
counter4        VAR     Byte
counter5        VAR     Byte
counter6        VAR     Word
pulse           VAR     Word
i               VAR     Byte
```

```
' Initialization ' ------------------------------------------------------------------------------
Initialize:   HIGH Reset                                        ' alert the DS1620
SHIFTOUT DQ, Clock, LSBFIRST, [WrCfg, %10]    ' use with CPU; free-run
LOW Reset
PAUSE 10
HIGH Reset
SHIFTOUT DQ, Clock, LSBFIRST, [StartC]                ' start conversions
LOW Reset
' -----------------------------------------------------------------------------
' Program Code ' ---------------------------------------------------------------------------
i=40
DO
PAUSE 1
LOOP UNTIL IN10=1

DO
i=i-1
GOSUB Temperature
DEBUG HOME,"Warming up for 40s.... : ",CR,
DEC2 i, "s",CR
PAUSE 1000
LOOP UNTIL i=0

LOW 5
  FOR counter = 0 TO 127
  PULSOUT 6, 1
  PAUSE 10
  NEXT

HIGH 5
  FOR counter = 0 TO 80
  PULSOUT 6, 1
  PAUSE 10
  NEXT
LOW 3
LOW 4
```

MAIN PROGRAM:

```
main:
DO
GOSUB Temperature
GOSUB Temp_Check
DEBUG CLS,HOME
DEBUG "DS1620", CR
DEBUG "------", CR
DEBUG SDEC tempC, " C    ", CR
DEBUG SDEC tempF, " F    ", CR

DEBUG CRSRXY,0,6,TAB, BIN1 IN0,
           TAB, BIN1 IN1,
            TAB,BIN1 IN2
PAUSE 100
IF IN0=1 AND IN1=0 THEN
  IF IN2=0 THEN
  GOSUB kill
  GOSUB position1
  ELSE
  DO
  GOSUB kill
  GOSUB position1
  PAUSE 3000
  GOSUB kill
  GOSUB position5
  PAUSE 3000
  LOOP UNTIL ((IN0=1) OR (IN1=1) OR (IN2=1))
  ENDIF
ELSEIF IN0=0 AND IN1=1 THEN
 IF IN2=0 THEN
 GOSUB kill
 GOSUB position3
 ELSE
 GOSUB kill
 GOSUB position4
 ENDIF
ELSEIF IN2=1 AND IN0=0 THEN
```

```
ELSEIF IN2=1 AND IN0=0 THEN
 IF IN1=0 THEN
 GOSUB kill
 GOSUB position5
 ENDIF
ELSEIF IN0=1 AND IN2=0 THEN
 IF IN1=1 THEN
 GOSUB kill
 GOSUB position2
 ENDIF
ELSEIF ((IN0=1) AND (IN1=1) AND (IN2=1)) THEN
 DO
 GOSUB swing
 LOOP UNTIL ((IN0=1) OR (IN1=1) OR (IN2=1))
ELSE
GOSUB kill
PAUSE 2000
ENDIF
LOOP
```

SUBROUTINES:

```
' ------------------------------------------------------------------------------
' Subroutines ' ----------------------------------------------------------------------------------
position1:
FOR counter1 = 1 TO 100
PULSOUT 15,1050
PAUSE 20
NEXT
RETURN

position2:
FOR counter2 = 1 TO 100
PULSOUT 15,900
PAUSE 20
NEXT
RETURN

position3:
FOR counter3 = 1 TO 100
PULSOUT 15,750
PAUSE 20
NEXT
RETURN

position4:
FOR counter4 = 1 TO 100
PULSOUT 15,600
PAUSE 20
NEXT
RETURN

position5:
FOR counter5 = 1 TO 100
PULSOUT 15,450
PAUSE 20
NEXT
RETURN
```

```
kill:
IF IN11=1 THEN
DO
PAUSE 1
LOOP UNTIL IN10=1
ENDIF
RETURN

swing:
pulse = 450

FOR counter6 =1 TO 120
PULSOUT 15,pulse
PAUSE 20
pulse=pulse+5
NEXT
PAUSE 1000
IF ((IN0=1) OR (IN1=1) OR (IN2=1)) THEN
GOTO main
ELSE
FOR counter6= 120 TO 1
PULSOUT 15,pulse
PAUSE 20
pulse=pulse-5
NEXT
PAUSE 1000
IF ((IN0=1) OR (IN1=1) OR (IN2=1)) THEN
GOTO main
ENDIF
ENDIF
RETURN
```

```
Temperature:                          ' read the DS1620
HIGH Reset                                ' alert the DS1620
SHIFTOUT DQ, Clock, LSBFIRST, [RdTmp]     ' give command to read temp
SHIFTIN DQ, Clock, LSBPRE, [tempIn\9]     ' read it in
LOW Reset                                 ' release the DS1620
tSign = sign                              ' save sign bit
tempIn = tempIn / 2                       ' round to whole degrees
IF (tSign = 0) THEN No_Neg1
tempIn = tempIn | $FF00                   ' extend sign bits for negative
No_Neg1:
tempC = tempIn                            ' save Celsius value
tempIn = tempIn */ $01CC                  ' multiply by 1.8
IF (tSign = 0) THEN No_Neg2               ' if negative, extend sign bits
tempIn = tempIn | $FF00
No_Neg2:
tempIn = tempIn + 32                      ' finish C -> F conversion
tempF = tempIn                            ' save Fahrenheit value
RETURN
Temp_Check:
IF tempC=tempC1 THEN
  PAUSE 1
  LOW 3
  LOW 4
ELSEIF tempC>tempC1 THEN
 HIGH 5
  FOR counter = 0 TO 20
  PULSOUT 6, 1
  PAUSE 10
  NEXT
  HIGH 3
  LOW 4
 ELSEIF tempC<tempC1 THEN
 LOW 5
  FOR counter = 0 TO 20
  PULSOUT 6, 1
  PAUSE 10
  NEXT
  LOW 3
  HIGH 4
ENDIF
PAUSE 1
tempC1=tempC                              ' pause between readings
RETURN
```

COST ANALYSIS

No.	Component	Quantity	Cost
1.	BASIC STAMP 2 Microcontroller	1	$106.00
2.	PIR (Passive Infrared) Sensor	2	$20.00
3.	PIR (Passive Infrared) Mini	1	$9.00
4.	Standard Parallax Servomotor	1	$11.00
5.	RadioShack Microfan	1	$10.00
6.	Temperature Sensor (DS1620)	1	$5.00
7.	Digital Potentiometer (AD5220)	1	$3.00
8.	Misc. (Pushbuttons, Transistor, Resistors, etc.)	-	$3.00
		Total	$167.00

Approximate Mass Production Cost = $70.00

FUTURE SCOPE

- Increase the range by incorporating a 360^o field of operation
- Increasing the precision by addition of multiple sensors
- Integration of the fan into a home automation system

Thank You.