Snow-Flake
The
Tank

By Paul Daignault
(and the grace of God,
and a little help from
Jeremy Coffey)

Table Of Contents

Abstract

I introduce to you, Snow-Flake the tank. Snow-Flake was a $300.00 venture in Paul Daignault's life. The purpose was to design an autonomous tank that fired disk projectiles at hot targets. The real purpose of Snow-Flake was to build a robot to attack my dog and keep him from annoying me. Both purposes have been realized, and 90% of all original goals have been reached. Original goals included an autonomous platform, heat sensors, bump sensors, ir sensors, and a firing gun. The only goal not obtained was a rotating turret. In the end, Snow-Flake's name should be changed to POJ-Bot (Piece-Of-Junk), because minimal amount of money and parts were used in the design. Yet I was amazed at the superior engineering that allowed Snow-Flake to become the robot that it is.

Executive Summary

Using State-of-the-Art technology, the 68HC11 allows break through capabilities in robot design. With the expansion ME11 overdrive board many analog and digital components were added to the robot.

IR sensors and bump sensors are at the heart of Snow-Flake. These sensors allow for the minimal collision avoidance, yet often leaving the robot stuck. The IR sensors give the robot eyes with about 120 degrees sight in the forward direction. The bump sensors are only placed in front of my robot. I could not get a bumper to stay connected. I tried super glue, epoxy, hot glue, tape, and a couple other strange ideas. Without the bumper the bump sensors have limited range and lead to my robot getting stuck a lot.

The heat sensor detects changes in heat. I was worried about getting a narrow field of view (accuracy) for the gun. So I simply built a barrel around the main heat sensor. This barrel worked surprisingly well with accuracy and heat sensing. The heat sensor and gun combination accurately shot me 15 feet away (when the batteries are fully charged).

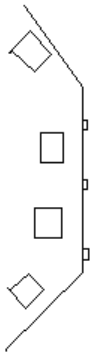
The gun uses two motors: one to propel the disks, and the other to pull the trigger. Both of these motors are powered by 9 volt batteries, and are controlled through 754410 motor driver chips. One motor spins rapidly and when the disk comes into contact it is propelled forward. The other motor pulls the trigger.

All four of these systems combine to make an autonomous Snow-Flake the tank robot system (Yippee!).

Introduction

This is Snow-Flake the Tank, an autonomous robot. It runs around and shoots heat sources. The third purpose of my robot is to be cool. Anything that fires is cool. Using all the systems already mentioned Snow-Flake behaves similar to a tank and is really cool. Its two in the morning, and not getting sleep is not cool.

IR and Bump Sensors

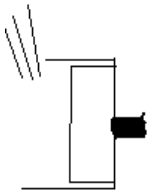


This is a picture of the front of my robot. The small squares are the bump sensors and the big squares are the IR sensors/emitters. Again the IR sensors give about 120 degrees of visibility.

Bump sensors have priority over IR sensors. Each sensor set has priority from left to right. For example, the left bump sensor has the highest collision avoidance priority in the robot (the heat sensor has the last).

The subroutines for both the IR and Bump sensors systematically check for each sensor and return a value other than 0 if they are set. If no change in the environment is detected then a 0 is returned. These values are passed to the motor controllor subroutine to determine how the robot reacts to the environment.

Heat Sensor



This is a pitiful drawing of my heat sensor. The barrel coming out of the front of the sensor is used to make the sensor be more accurate. This is actually used to aim the gun. When the heat sensor trips the gun is aiming at the target.

The three wires hanging out of the bottom of the sensor are +5, gnd, and the analog signal. The analog signal has a nominal value of about 127 and range of anyting between 0 and 255. The heat sensor subroutine will return a heat detected signal when the range is outside of 140-114. This is passed to the motor controllor subroutine.

Motor Controllor

The motor controllor received values from the bump sensor, IR sensors, and Heat sensor. The priority is given in the respective order. When a bump sensor or IR sensor is tripped the robot reverses and turns away for the sensor that was tripped. The only difference between the IR and bump sensor motor controllor routine is how long the robot reverses for.

When the motor controllor receives a heat sensor tripped flag, the robot stops and fires the gun. I wish the gun would shoot me! Its 3 in the morning and I'm drooling on the keyboard.

Gun Controllor

You don't want me to try and draw a picture for this one. Basically the gun has two motors controlled by two 754410 motor driver chips. These chips are controlled by the 68HC11,a 753 output chip, and two 9 volt batteries. The first motor propels the disks. Its a low torque, high speed motor. The disk is forced into contact with the spinning motor and is fired forward. I designed the 754410 chip to turn the motor on when enabled.

The second motor is connected to a spring mounted trigger on the gun. The motor turns to pull the trigger. The motor will torque out when the trigger is full pulled. After a short amount of time the motor turns off. When its time to fire, the motor loosens up the trigger (the spring will make it return to reload position) and then pulls the trigger to firing range. The 754410 chip was designed similar to the motor drivers for the ME11 (thanks Scott!).

ME11 and 68HC11

These two components are the brains of the robot. The ME11 has two motor drivers that are connected wheels. These two wheels control the robots motion using directions from the motor controllor subroutine. The 68HC11 has 4 Analog to digital convertors. These four converters are used with the 4 IR sensors and the heat sensor. All this digital data is used to examine the outside environment and make modifications to the motors or gun.

Behaviors

The robot acts like a tank. At the moment it runs around randomly, and when it finds a heat source it shoots at it. A standard hostile tank has been developed. If more time was available I would have developed a good moving algorithm instead a using a random one. Two emergent behaviors are: My dogs runs like hell to get away, and kids cry when they get shot.

Conclusion

This is one smooth robot. It fires disks which automatically adds it to the cool robot hall of fame. Actually I spent sometime on the code getting the robot to be smooth and not jerk around a lot. I wish I had a little more time to get the robot to be more life like. I enjoyed making my POJ bot, it was a great thrill to actually have it fire the disks. Jeremy and I had a lot of fun cracking jokes and making fun of each others robots as we built them (His really is a piece of junk!).

```
int priority_detect;
float mone = 45.0;      /*typical forward motor*/
float mzero = 55.0;     /*speeds              */

void main()
{
init();               /*test robot parts*/
robot_forward();      /*initial motor start*/

while (16 & peek(0x4000)) /*while switch is on do this loop*/
{
priority_detect = 0;

if (priority_detect == 0)
        priority_detect = bump_setup();
if (priority_detect == 0)
        priority_detect = ir_setup();
if (priority_detect == 0)
        priority_detect = heat_setup();
if (priority_detect != 0)
        motor_cont();
}
hard_stop();          /*switch off? turn off motors*/
}



/*determines if any sensors are tripped, and if
  so, how to deal with them.  mostly used for obstacle
  avoidance.  The heat sensor section fires gun.*/
void motor_cont()
{
int x,y;
int temp;

/*This chunk of code is used for Bump and IR sensors*/
/*establish reverse time based upon bump or IR sensors*/
if (priority_detect < 4)
{
        hard_stop();
        y = 2500;
}
else
{
        soft_stop_forward();
```

```
        y = 1500;
}

/*motor controllor for Bump and IR Sensor flags*/
if (priority_detect < 13)
{
        for (x=0;x<500;x++);
        robot_reverse();        /*go_backwards*/
        for (x=0;x<y;x++);
        soft_stop_reverse();
        if (priority_detect == 1 || priority_detect == 10)
                turn_mzero();
          else                  /*turn direction*/
                turn_mone();
        y = peek(0x100e) * 5;  /*random angle to turn to*/
        if (y>600)
          y = 600;
        for (x=0;x<y;x++);
        if (priority_detect == 1 || priority_detect == 10)
                soft_stop_mzero();
          else
                soft_stop_mone();
        robot_forward();        /*go_forward*/
        return;
}
/*This chunk of code deals with the heat sensor, and
  firing the gun*/
else if (priority_detect == 20)
{
        soft_stop_forward();
        fire_gun();
        robot_forward();
}
return;
}


void soft_stop_mzero()
{
int x,y;                /*soft_stop*/

for (x=(int)mone;x>=0;x--)
{
   for (y=0;y<4;y++);
   motor(1,-(float)x);
```

```
   motor(0,(float)(10+x));
}
hard_stop(); /*stops both motors*/
}

void soft_stop_mone()
{
int x,y;              /*soft_stop*/

for (x=(int)mone;x>=0;x--)
{
   for (y=0;y<4;y++);
   motor(1,(float)x);
   motor(0,-(float)(10+x));
}
hard_stop(); /*stops both motors*/
}

void soft_stop_reverse()
{
int x,y;              /*soft_stop*/

for (x=-(int)mone;x<=0;x++)
{
   for (y=0;y<5;y++);
   motor(1,(float)x);
   motor(0,(float)(-10+x));
}
hard_stop(); /*stops both motors*/
}

/*when moving forward, softly stops the robot
  instead of a complete stop (hard_stop)*/
void soft_stop_forward()
{
int x,y;              /*soft_stop*/

for (x=(int)mone;x>=0;x--)
{
   for (y=0;y<5;y++);
   motor(1,(float)x);
   motor(0,(float)(10+x));
}
hard_stop(); /*stops both motors*/
}
```

```
void robot_reverse()
{
int x,y;              /*reverse*/

for (x=0;x>=-(int)mone;x--)
{
  for (y=0;y<5;y++);
  motor(1,(float)x);
  motor(0,(float)(-10+x));
}
}

void robot_forward()
{
int x,y;              /*forward*/

for (x=0;x<(int)mone;x++)
{
  for (y=0;y<5;y++);
  motor(1,(float)x);
  motor(0,(float)(10+x));
}
}

void turn_mzero()
{
int x,y;              /*turn*/

for (x=0;x<(int)mone;x++)
{
  for (y=0;y<4;y++);
  motor(1,-(float)x);
  motor(0,(float)(10+x));
}
}

void turn_mone()
{
int x,y;              /*forward*/

for (x=0;x<(int)mone;x++)
{
  for (y=0;y<4;y++);
```

```
   motor(1,(float)x);
   motor(0,-(float)(10+x));
}
}

void hard_stop()
{
motor(1,00.0);          /*stop*/
motor(0,00.0);
}

void fire_gun()
{
int x;

poke(0x4000,0b11100000); /*Reload*/
for (x=0;x<300;x++);

poke(0x4000,0b10000000); /*let engine*/
for (x=0;x<1500;x++);     /*warm up   */

poke(0x4000,0b11010000); /*Pull trigger*/
for (x=0;x<2000;x++);

poke(0x4000,0b00000000); /*Shut down gun*/
}


/* checks the heat sensor for any variations in
   temperature.  if So, it returns a 20 */
int heat_setup()
{
int temp;

temp = analog(3);

if (temp > 140 || temp < 114)
   return 20;
else
   return 0;
}

int ir_setup()
{
```

```
int sens4 = 0;
int sens3 = 0;
int sens2 = 0;
int sens1 = 0;

if (analog(4) > 92)
        sens4 = 1;
/*removed for use by the heat sensor*/
/*if (analog(3) > 92) */
        sens3 = 0;
if (analog(2) > 92)
        sens2 = 1;
if (analog(1) > 92)
        sens1 = 1;

/*these four if statements determine which
  direction the IR detection is coming from
  and returns the appropriate flag to the
  motor controllor through priority_detect*/
if (!sens1 && !sens2 && !sens3 && !sens4)
        return 0;
if (!sens1 && !sens2 && !sens3 && sens4)
        return 10;
if (sens1 && !sens2 && !sens3 && !sens4)
        return 11;
if (sens2 || sens3)
        return 12;
return 0;
}


/*detects which bump sensors are set and
  returns an appropriate value back to
  motor controllor through priority_detect*/
int bump_setup()
{
if (peek(0x4000) < 32)
        return 0;
if (128 & peek(0x4000))
        return 1;
if (32 & peek(0x4000))
        return 2;
if (64 & peek(0x4000))
        return 3;
return 0;
```

```
}


/*Does an initial tests to see if things
  are working in snow-flake*/
void init()
{
int x;

poke(0x7000,0xff); /*turn on IR emitters*/
poke(0x4000,0x00); /*clears up garbage on 4000*/

/*Gun Test*/
for (x=0;x<2000;x++);
poke(0x4000,0b11010000); /*trigger test*/
for (x=0;x<2000;x++);

poke(0x4000,0b10000000); /*gun engine*/
for (x=0;x<500;x++);     /*test      */

poke(0x4000,0b00000000); /*Shut down gun*/

/*Motor Test*/
motor(1,-mone);         /*turn*/
motor(0,mzero);
for(x=0;x<500;x++);
hard_stop();            /*stop*/
for(x=0;x<200;x++);
motor(1,mone);          /*turn*/
motor(0,-mzero);
for(x=0;x<500;x++);
hard_stop();            /*stop*/
for (x=0;x<2000;x++);
}
```