University of Florida
Department of Electrical Engineering
EEL 5934
Intelligent Machines Design Laboratory
Professor Keith L. Doty

Erin II, The Predator

By
Sherman Butler

12/4/95

TABLE OF CONTENTS

ABSTRACT

For my antonymous robot, I chose a predator based design. I focused on three distinct predator features: speed, maneuverability, heightened sense of environment, and the ability to sense it's prey.

I used a circular platform for the robot to maximize it's maneuverability. Also, I used two modified Futaba servos for the drive-train which generate a lot of torque, but they are slow. Therefore, I used larger airplane wheels, 3 1/2" in diameter, and a higher voltage power supply (12 AA cells) to increase the robots speed.

The Futaba servo are not precision motors, and there is a rotational speed discrepancy between the two servos which causes the robot to slowly turn to one side instead of moving in a straight line. I added wheel encoders to compensate for this speed discrepancy. The wheel encoders are based on commutator noise sensors whose signals are decoded by the HC11's input-capture ports.

I installed I/R transmitters and receivers for collision avoidance to the robot's platform. These sensors are adequate for highly reflective environments but fail when confronted with dark non-reflective obstacles. I added a collision sensor or bumper to the robot to allow it to sense non-reflective obstacles. The bumper is positioned at the front of the robot, and it uses two lever switches and a resistor voltage divider tree to generate a 4 level analog voltage signal which is decoded by the HC11's A/D port.

I designated 40Khz infrared as the robot prey smell which can be detected by the Sharp infrared sensors, and I installed additional Sharp infrared sensors to enable the predator robot to sense it's prey from any direction. Also, I installed two Cadmium-Sulfide light sensor for light following which will be used to enhance the robot's mode of operation.

EXECUTIVE SUMMARY

I designed and built an autonomies predator robot. The robot is based on a circular platform which has a diameter of 11.5 inches. The drive train uses two modified Futaba servos, and large, 3.5”, airplane wheels. The power supply is provided by twelve AA batteries in series.

The 68HC11EVBU board is used for the robot’s CPU and prototype circuit board. The microprocessor has 32K of expanded memory which is mapped into it’s upper memory map and an external memory mapped digital output latch attached to the microprocessor’s external bus. The digital latch is used to drive the external infrared transmitters.

The robot has a total of ten individual sensors which can be broken into four types of sensors. Two wheel encoder sensors are attached to the HC11’s input-capture port. Five Sharp infrared sensors and two cadmium-sulfide light sensors are connected to the HC11’s A/D ports. Four of the Sharp I/R sensors have I/R transmitters for collision avoidance and wall following. The fifth I/R sensor is located at the rear of the platform, and it is used to sense other robot I/R.

Finally, a contact sensor or bumper is located at the front of the robot, and is attached to the HC11’s A/D port. The bumper has two contact quadrants which enable the robot to sense which side of the robot the obstacle is on. The bumper is also used to detect capture of it’s robot prey.

INTRODUCTION

A predator can sense, pursue, and capture it's prey. With this in mind, I set the scope of my robot design to mimic these features. But first I needed to answer the following questions. How does the predator sense it's prey? How does the predator pursue it's prey? How does the predator capture it's prey ?

Predators sense their prey with enhanced senses which focus on a distinct smell of the prey. In the robot world, this translate to modulated infrared. Therefore, my robot must sense the other robots modulated infrared while using it's own modulated infrared for obstacle avoidance. This will require the robot to halt it's forward motion and disable it's infrared transmitter while sensing the other robot's infrared. This is easily done, and it is used for my robot.

Once the predator senses it's prey, what does the predator robot do next? The predator robot pursues it's prey by tracking the prey's infrared signal using it's front two infrared sensors while it's infrared transmitters are intermittently turned off. This does not allow the predator robot to sense the other robot's motion, and raises a question as to how the robot avoids the potential of crashing into another robot or a non-reflective surface? This problem dictates the need for additional sensors such as a radar, sonar, or a collision sensor.

An in-expensive and simple Radar is not currently available for the general public except possibly radar-on-a-chip which isn't currently available. Radar-on-a-chip won't be available until long after I have completed this class. Also, the complexity and price of sonar out-weigh it's benefits. Therefore, I chose a collision sensor or bumper for my robot. A bumper is simple and easily to implemented, and it is guaranteed to detect any standing obstacle.

. How does the predator capture it's prey. For my robot design, I settled on a simple scheme. Robot platforms do not provide an ideal reflective surface which I/R collision sensor will be able to detect. Therefore, a planned or unplanned a collision is likely to occur when two or more robots are placed in the same environment. Therefore, the predator robot will seek out external infrared sources, and it will use it's bumper to detect a collision a with another robot. When the predator robot senses a collision with an external object which is emitting 40Khz infrared, it concludes that it captured another robot prey.

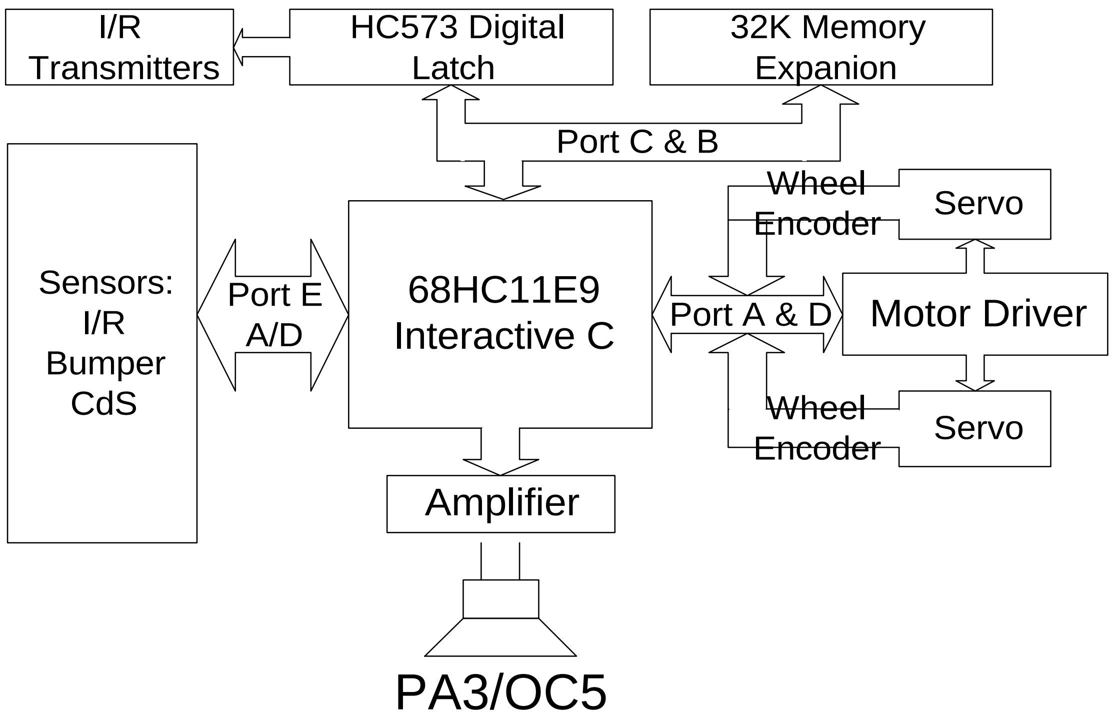
This scheme isn't full-proof, and it will fail when trying to detect a large reflective robot. Because the predator's own infrared collision sensor will keep it from making contact with the reflective robot. Also, if the predator makes contact with a wall, and senses reflected infrared from another robot, it will make a false capture. There is also the potential of making contact with the prey robot while not sensing the prey's infrared which would cause the predator to discard the captured prey. The technique has it's down-falls, but it is simple and it will work in most situations.

MAIN BODY

Integrated System

The predator robot will be based on the 68HC11EVBU board which uses the ‘E9 chip. The microprocessor is used in expanded mode with 32K of external ram which is mapped into the upper memory map. The robot’s functions are programmed in C using freeware Interactive C. The robot's general block diagram is depicted in Figure 1. Note, the behavior diagrams are located in the behavior section.

Figure 1.



Mobile Platform

In order to catch it's prey, the predator must be more agile and faster. In the robot world, this translate to speed and maneuverability. Traditional four wheel platform have a large turning radius which limits it's abilities to make a turn in tight spaces. It must make a multiple point turn to escape a corner, and it might get trapped.

A circular shaped platform enables a robot to make a 360° turn in a tight corner. Also, it is less likely to get hung up on a sharp edge or corner. Therefore, a circular platform was chosen for the robot platform which was cut from plywood.

A good lesson can be learned from the wood used for the platform. On the advice of a TA, I purchased birch 4-ply aircraft plywood which is very expensive. Another more resourceful student purchased 3-ply birch wood at 1/9 the cost per square inch at a hardware store. It pays to be more resourceful and explore your options.

For the drive-train the traditional Futaba servo were chosen due to it's simplicity, price, and durability. But, the traditional robot wheel size, which is 2 3/4", were not used. In order to meet the speed requirement, larger wheels were chosen for my robot. I settled for 3 1/2" wheels which I purchased from the hobby store. The large wheels should provide a 27% increase in speed. Also, I increased the robot’s power supply to twelve AA batteries. But, I might reduce the number of batteries to eight, because twelve batteries make the robot to fast and uncontrollable, and the batteries dissipate their charge more rapidly at the higher voltage.

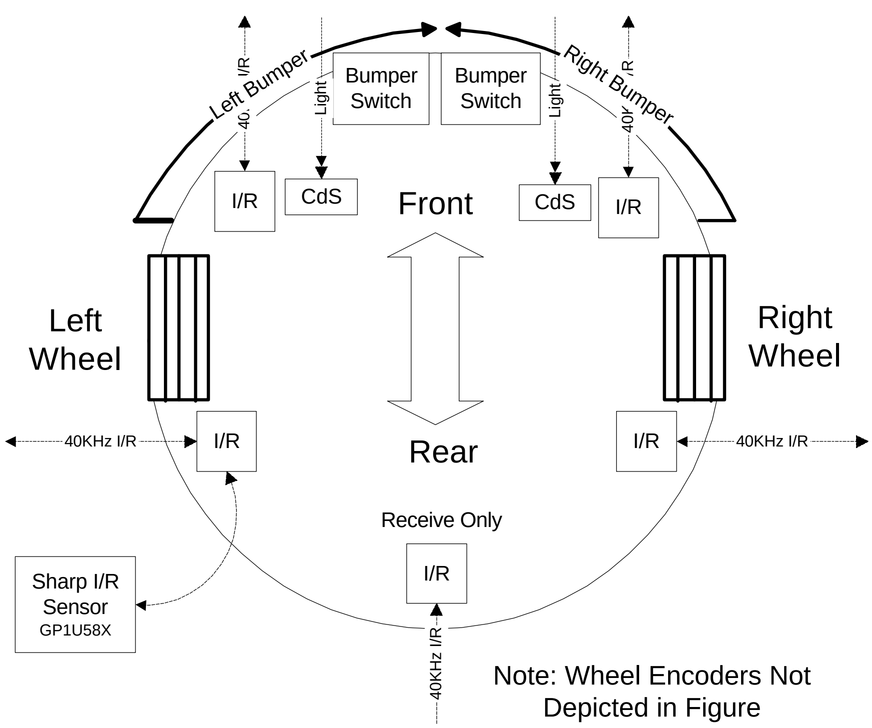
Sensors

The predator robot must be capable of sensing it's prey and it's environment. If robots are it's prey, then modulated infrared is their most distinct ‘smell’. Also, a predator must capture it's prey's while avoiding obstacles. With these requirements, I have settled on four primary sensors which are summarized in table 1. Also, the sensor layout is depicted in figure 2.

Table 1: Sensor Summary

Type	Qt.	HC11 Port	Function
Sharp I/R GP1U58X	5	Port E (A/D)	Infrared sensor and Collision Avoidance
Collision Sensor	2	Port E (A/D)	Front Left and Right Collision Sensor
Cadmium-Sulfide	2	Port E (A/D)	Light Sensor
Wheel Encoder	2	Port A (IC)	Commutator Noise Wheel Encoder

Figure 2: Sensor Layout.



Infrared

The infrared sensors are used for two functions, collision avoidance and prey detection. The I/R sensor is provided by a modified Sharp, GP1U58X, infrared sensor which is capable of detecting 40Khz I/R. The digital sensor is converted to an analog device by bypassing it's digital circuitry while re-routing it's internal analog voltage to the HC11's A/D port. The sensor's nominal analog voltage is 1.5V which increases to a maximum 2.5V with I/R saturation.

To improve the sensors resolution, adjustable external resistors are connected to the HC11's voltage reference low and high. The references were adjusted to maximize the sharp sensors resolution without degrading the performance of the other sensors.

Collision avoidance is accomplished by combining an infrared emitting diodes with the sharp sensor. The diode is pointed away from the sharp sensor, and it is modulated by a 40Khz signal which is generated by a counter and the E-clock. When the I/R light is reflects from an object, the sharp sensor will detect the reflected I/R. The infrared LEDs are driven by a HC573 latch which is attached to the external memory bus. This enable range finding abilities and the ability to turn the I/R diodes off for prey detection.

As depicted in figure 2, the front four I/R sensor, which have I/R LED's attached to their cases, are used for collision avoidance. The rear sensor has no I/R transmitter, and the sensor is used only to detect the presence of prey robots located in it's the rear quadrant. When the I/R Leeds are turned off, the front four I/R sensors are also used to detect the prey modulated I/R.

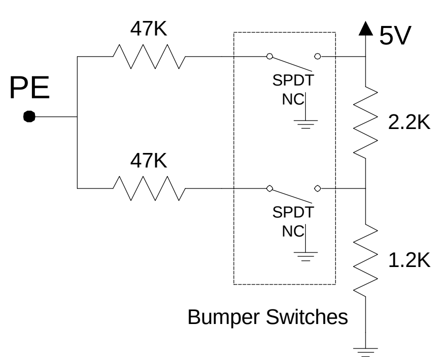
Collision Sensor

A collision sensor is used to halt the predator's attach and sense non-reflective objects which the I/R collision sensors failed to detect. The collision sensor is a bumper based on the bumper design introduced to the class by Joel who is another student in the class.

The design consist of two pieces of coat-hangers mounted to each side of the front of the robot. The rigid wires are connected to a modified roller switch. The rollers of the switch are removed, and it’s frame is used to connect the bumper wire to the switch. When a collision occurs, the bumper wire activates the roller switches.

The bumper switches generate a digital signal, but there are no available digital inputs. Therefore, I am using a voltage divider circuit introduced in Mobile Robots, by Jones and Flynn on pg 279 to connect the two digital switches to the A/D port. The circuit is depicted in figure 3.

Figure 3: Collision Sensor Circuit

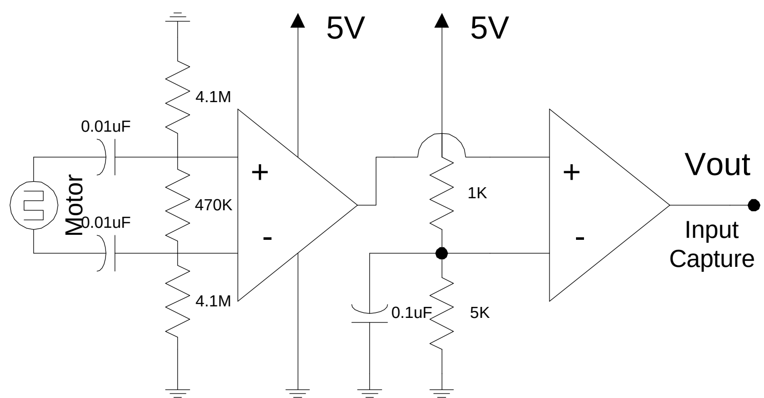


If the circuit introduced in the class had provided digital inputs or a port expansion, I wouldn’t have to use the A/D port for a digital input.

Wheel Encoders

A commutator noise sensor was introduced in class by TA Eric, and I incorporated it into my design. The sensor uses a LM324 Op-amp as a noise amplifier and a comparator. The final circuit configuration is depicted in figure 4, and the design was derived from experimentation.

Figure 4: Commutator Noise Sensor



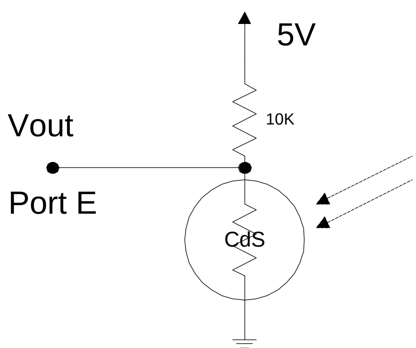
There are two op-amp stages used for each noise sensor. The first op-amp stage converts the noise to a square wave with minor noise, and the second op-amp is used as a comparator which removes the majority of remaining noise from the waveform. The circuit produces a square wave output.

Each servo is directly connect to the noise sensor at the H-bridge, and the conditioned output from the noise sensor is connected to the HC11's input capture 1 & 2. I have received poor results when sampling the pulse train in IC which can be contributed to IC's lack of timing or consistency. Therefore, I have not included a speed routine to my robot's behavior. I would prefer to explore the potential of the speed routine at a latter time.

<u>Cadmium-Sulfide</u>

I purchased an assortment of cadmium sulfate sensors from Radio Shack for light detection, and I have connected two of the CdS sensors to the A/D input with a pull-up resistor. The circuit was taken from <u>Mobile Robots</u>, by Jones and Flynn on pg 279, and the schematics is shown in figure 5.

<u>Figure 5: Cadmium-Sulfide Light Sensor</u>



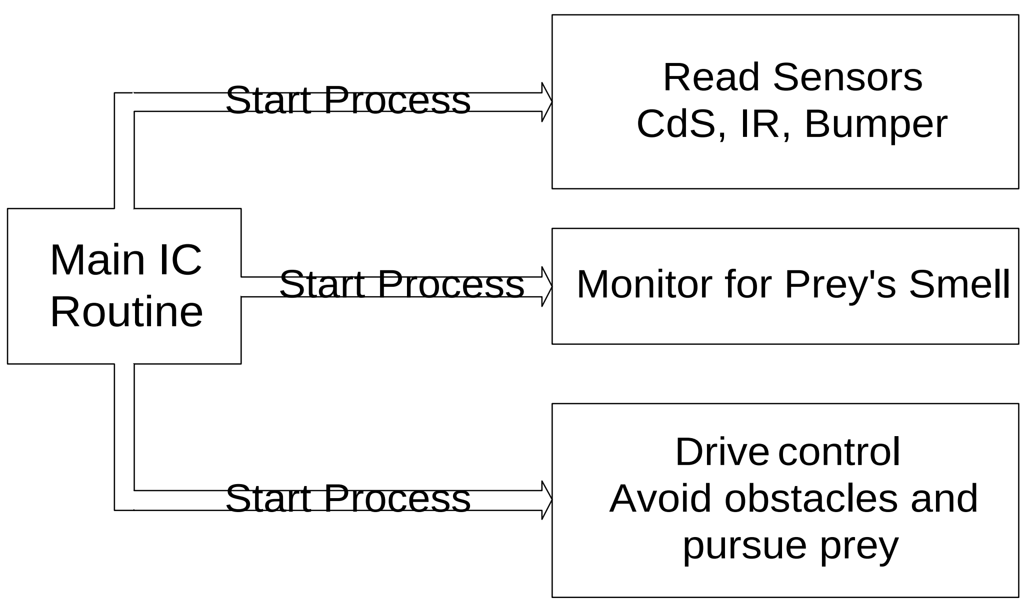
I found the cadmium-sulfide light sensors to be extremely sensitive, and I currently have two of the sensors connected to the front of the robot. The light sensors are used for light following.

Behaviors

For my predator robot, I have attempted to implement two primary modes of operation, wandering and pursuing. When the robot is wandering, it systematically moves around the room while avoiding obstacles. Also, it occasionally smells it's environment for other robot prey. When the predator robot senses another robot, it enters pursuit mode. In this mode, it attempts to tracking the prey with it's five sharp I/R sensors. When the robot makes a collision while sensing modulated I/R, it will assume it has captured it's prey. This presents the potential of false capture when the predator runs into a wall while sensing reflected I/R. This is acceptable because a predator doesn't always capture it's prey in nature.

To implement the predator behavior, I used three processes in IC which are shown in figure 6.

Figure 6: Predator Robot Processes



The ‘Read sensor’ process reads the eight sensors attached to the A/D port. First, it gets the current bumper status and the five sharp infrared sensors while the infrared diodes are disabled. Then, the process enables the infrared diodes and records the I/R readings after the sensors have stabilized. The cadmium-sulfide sensors are also read, but their value are not used for the current behavior routine. I plan to implement a behavior feature in the future.

The ‘monitor for prey smell’ process compares the recorded I/R values ,when the I/R diodes are disabled, to a threshold value which dictates a prey in the vicinity. If another robot’s I/R is sensed, the process sets a flag which the ‘drive control’ process monitors.

The ‘drive control’ process monitors the status of the collision sensors and the prey status flag. If no collision has occurred and the prey flag is set, the process attempts to maneuver the robot in the general direction of the prey. If a collision occurs when the prey flag is set, the predator has either captured it’s prey or abandon the pursuit due to an obstacle. Also, if no prey is sensed, the ‘drive control’ process maintains the robots current path until a prey is smelled or and obstacle is encountered. The ‘drive control‘ process implements the wander and pursuit behavior of the predator robot.

CONCLUSION

The hardware half of the robot project is complete, and it is working as expected. The predator robot behavior routines are simple and work to a certain extent. But the abilities of the robot have yet to be tapped. Due to time restrictions, the behavior potentials of the robot have not been fully tapped. It would appear that the extension class is the only chance to truly explore the potential of the robot platform.

If the robots were prefabricated or sold in a kit form, I could have explored the behavior potentials of the robot more instead of spending the whole semester assembling the robot. Don't get me wrong. I enjoyed assembling the robot, and I am sure it will make a nice paper weight in the future. I just would of liked to have more time to tap the potential of the robot.

With this in mind, I still find myself regretting taking this class so late in my academic career. This is my last term, and I have to shuffle my senior design project (largest time consumer), job interviews (annoying), current class load, and my personal life (which I don't have much of) which includes a baby. These factors take the fun out of this class.

This class is meant to be fun, and it provides the student a chance to build something from scratch and tinker with it till your heart is content. I wish I had taken this class two semester ago when I had more time and could have enjoyed it more.

Overall, I enjoyed the class.

APPENDIX A: PART LIST

Part	Qt.	Price	Total Price	Retailer
74HC573 or 373	2	1.99	3.98	JDR
74HC138	2	.99	1.98	JDR
74HC14	1	.99	.99	JDR
74HC390	1	.99	.99	IML
L293NE	1	4.00	4.00	IML
LH52256	1	4.99	4.99	JDR
GP1U58X	4	3.00	12.00	Dr. Doty
Cadmium-Sulfide	2	1.99	3.98	Radio Shack 276-1657
I/R LED	4	1.99	7.96	Radio Shack 276-143
Futaba Servos	2	15.00	30.00	Dr. Doty, FP-S148
3.5” Wheels	2	5.00	10.00	Hobby Land
Micro-lever Switch	2	1.99	3.98	Radio Shack
1.5” Caster	1	2.99	2.99	Zells Hardware
4-cell AA holder	3	1.99	5.97	Radio Shack
4-pack AA Batteries	3	4.99	14.97	Wal-mart
Wire-wrap Sockets	9		10.00	
Birch Plywood	1		10.00	Paid to much
			$128.78	Total Cost

Note: part list doesn’t include EVBU board and miscellaneous items such as: wire, resistors, capacitors, wire-wrap Ids, and connectors.

APPENDIX B: BEHAVIOR SOFTWARE

```
/* Sherman Butler */
/* Erin II, The Predator */

int RETREAT_TIME = 750, TURN_TIME=750, Cool_down_time=1000;
int IR_Threshold = 160;
int STOP =       0;
int FORWARD =    1;
int BACKWARD =   2;
int LEFT_TURN =  3;
int RIGHT_TURN = 4;
int LEFT_ARC =   5;
int RIGHT_ARC =  6;
int NORM_SPEED = 50;
int LEFT_MOTOR = 0;
int RIGHT_MOTOR= 1;
int NO_Contact=240, Left_Contact=200, Right_Contact=15;

/* Wheel Encoders */
int right_clicks, left_clicks, diff;

/* External Sensor states */
int Bumper, Left_CdS, Right_CdS;

/* IR Sensor values with diodes turned off */
int Left_front, Right_front, Left_rear, Right_rear, Rear_IR;

/* IR Sensor values with diodes turned on */
int Left_front_IR, Right_front_IR, Left_rear_IR, Right_rear_IR;



void init()
{
  poke(0x1009, 0b00111110); /* Set DD[5-4] to outputs     */
  poke(0x1028, 0);          /* Disable SPI Port's Wired-OR*/
}

void wait(int milli_seconds)
{
  long timer_a;
  timer_a = mseconds() + (long) milli_seconds;
  while(timer_a > mseconds())
    {
    defer();
    }
}


void move(int operation)
{
  if (operation == STOP) ao();

  else if (operation == FORWARD)
    { motor(LEFT_MOTOR, NORM_SPEED+20);
      motor(RIGHT_MOTOR,NORM_SPEED); }
```

```
  else if (operation == BACKWARD)
    { motor(LEFT_MOTOR, -(NORM_SPEED+20));
      motor(RIGHT_MOTOR,-NORM_SPEED); }

  else if (operation == LEFT_TURN)
    { motor(LEFT_MOTOR, -(NORM_SPEED+20));
      motor(RIGHT_MOTOR,NORM_SPEED); }

  else if (operation == RIGHT_TURN)
    { motor(LEFT_MOTOR, NORM_SPEED);
      motor(RIGHT_MOTOR,-NORM_SPEED); }

  else if (operation == LEFT_ARC)
    { motor(LEFT_MOTOR, STOP);
      motor(RIGHT_MOTOR,NORM_SPEED); }

  else if (operation == RIGHT_ARC)
    { motor(LEFT_MOTOR, NORM_SPEED+20);
      motor(RIGHT_MOTOR,STOP); }
}

void escape(int TURN)
{
 move(BACKWARD);
 wait(RETREAT_TIME);
 ao();
 wait(Cool_down_time);
 move(TURN);
 wait(TURN_TIME);
}

void rotate()
{ move(RIGHT_TURN);
  wait(2*TURN_TIME);
}

void turn_right()
{
  move(RIGHT_TURN)
  wait(TURN_TIME);
}

void turn_left()
{
  move(LEFT_TURN)
  wait(TURN_TIME);
}

void arc_right(ARC_TIME)
{
  move(RIGHT_ARC)
  wait(ARC_TIME);
}

void arc_left(ARC_TIME)
{
  move(LEFT_ARC)
  wait(ARC_TIME);
}
```

```
void read_wheel_encoders()
{
 while(1)
  { enable_encoder(0);
    enable_encoder(1);
    wait(100);
    right_clicks=read_encoder(0);
    left_clicks=read_encoder(1);
    disable_encoder(0);
    disable_encoder(1);
    diff=right_clicks - left_clicks;
    defer();
  }
}




void read_sensors()
{
    while(1) {
    /* Read IR sensors with LED turned off */
        wait(10); /* wait for the IR to stabilize*/
        Left_front  = analog(0);
        Right_front = analog(1);
        Left_rear   = analog(3);
        Right_rear  = analog(5);
        Rear_IR     = analog(4);

        poke(0x4000, 0xff);     /* Turn on IR LEDs */
        wait(10); /* wait for the IR to stabilize*/

        /* Read non-IR related sensors */
        Bumper         = analog(2);
        Left_CdS       = analog(6);
        Right_CdS      = analog(7);

        /* Check these sensors last to allow them to stabilize */
        Left_front_IR  = analog(0);
        Right_front_IR = analog(1);
        Left_rear_IR   = analog(3);
        Right_rear_IR  = analog(5);
        poke(0x4000, 0x00);     /* Turn off IR LEDs */
        defer();
    }
}

void monitor_prey()     /* Monitor environment for robot smell */
{ while(1)
  {
  if ((Right_front>Prey_threshold)||(Left_front>Prey_threshold)||
      (Right_rear>Prey_threshold)||(Left_rear>Prey_threshold)||
      (Rear_IR>Prey_threshold))
      Prey=1;
  else
      Prey=0;
  sleep(1.0);
```

```
  defer();
  }
}


void drive_control()
{ while(1)

 {
  if (Bumper>NO_Contact)      /* Collision Hasn't occured */

  {
    if (Left_front_IR>IR_Threshold && Right_front_IR <= IR_Threshold)
        move(RIGHT_ARC);
     else if (Right_front_IR > IR_Threshold && Left_front_IR <=
IR_Threshold)
        move(LEFT_ARC);
     else if (Left_front_IR > IR_Threshold && Right_front_IR >
IR_Threshold)
        escape(RIGHT_TURN);
     else
        if (Prey)  /* Was a prey sensed */
         {                      /* Pursue the prey */
         if (Rear>Prey_threshold)  /* Prey in rear */
            rotate();
         else if (Right_rear>Prey_threshold) /* Prey to the right */
            turn_right();
         else if (Left_rear>Prey_threshold)  /* Prey to the Left */
            turn_left();
         else if (Left_front>Right_front)    /* Prey is to the Left
Front*/
            arc_left((Left_front-Right_front));
         else if (left_front<Right_front)    /* Prey is to the Right
Front*/
            arc_right((Right_front-Left_front));
         else
            move(FORWARD);   /* Else, Prey is directly in front */
         }
   }
   else     /* Else, A collision has occured */
   {
     if (Prey)
        Prey=0;                  /* Prey Was Captured */
/* Check location of obstacle and retreat */
     if (Bumper>Left_Contact)   /* Obstacle on left side of robot*/
        escape(RIGHT_TURN);
     else if (Bumper>Right_Contact)  /* Obstacle on right side of
robot*/
        escape(LEFT_TURN);
     else                           /* Obstacle in front of robot */
        escape(RIGHT_TURN);
   }
   defer();
 }
}


void main()
{
  init();
```

```
  start_process(read_sensors());
/*  start_process(read_wheel_encoders()); Removed for now */
  start_process(drive_control());
  start_process(monitor_prey());
}
```