EEL 5666
Intelligent Machine Design Laboratory
Professor Keith Doty

Project

MINE SWEEPER

By **Nick H. Nguyen**

Electrical and Computer Engineering Department
University of Florida

TABLE OF CONTENTS

Abstract

The "Mine Sweeper" is a mine deactivating robot. It is supported to handle the job by itself without remote controlling. The robot is supposed to detect and deactivate four IR mines at four corners of the mine field; avoid four red mines and one magnetic mine, then return to its starting location by following a flash light.

The mine field is 5'x6'3/4 flat white surface. A 5" white wall surround the playing surface. There are four fixed mines that can be recognized by 6" red square painted around the mine and they are located half way from a corner to the center of the mine field. Also there is one random magnetic mine. The IR transmitters will be located 3" above the playing surface.

A Motorola 68HC11 EVBU board with E9 version MCU and Novasoft ME11 expansion board is the robot's integrated system. The M68HC11-E9 is a 8-bit MCU which has many advanced features for this project. A Talrik Junior (TJ) platform, which is produced by Novasoft, is suitable for this project because of its round shape (6 1/2 inches in diameter) and light weight. The sensory system includes some different types of sensors, such as: Bumper sensor, IR sensor, Magnetic Field sensor, and Photodiode.

Introduction

Today, there are millions of acres of mine fields around the world, and thousands of innocent people are killed and injured every year. To minimize human loss, an autonomous machine can be designed to deactivate the mine fields by itself. The robot can be intelligent enough to handle the job without remote control from a human. This project is not the first attempt in this field; however, my primary goal is designing an inexpensive, efficiency, and reliable robot, which can deactivate a mine field in the shortest time and consume the least energy.

My robot will use a Motorola 68HC11-E9 microcontroller. A TJ platform is suitable for this project. There are six integrated behaviors that use 4 types of sensors. The robot will look for IR light, which is emitted from one of four mines on four corners, and deactivate the mine. It will also know how to avoid magnetic mines and some other obstacles.

Specifications

- System: self operating
- IR detector range: 10 feet
- Magnetic sensor sensitivity: 5” to 8” from a 1” diameter coil (5.5 kHz)
- Red Color Detector Response Time: less than 100□s (14 kHz)
- Operating Time: 20 minutes (estimated at full charged)
- Size: 7” in diameter; 6” in height
- Power Supply: 2 packs of 6-AA NiCd batteries
- Require: a PC (DOS and IC) with a serial port, and a RS232 serial cable
- Detect red and white color using photodiodes

Integrated System

The Mine Sweeper uses a Motorola M68HC11-E9 EVBU micro controller board which is easy to program and the price is low. Also, it can do many different functions, such as: Real Time Interrupt (RTI), Input Compare (IC), Output Compare (OC), Pulse Accumulator (PA), and Analog to Digital Converter (A/D).

A ME11 expansion board (produced by Novasoft) is added to the EVBU. The ME11 is designed especially for robotic applications. Its features include:

- 32Kbytes of RAM.
- 8 bits of digital output capable of driving a total of 75ma continuous current.
- A single output can drive as much as 35ma of continuous current.
- 4 digital input and 4 digital output enables controlled by the R/W line and E-clock of the MC68HC11 processor.
- An H-bridge motor driver with thermal shutoff that can drive two small DC motors with maximum sustained current of 1A through each motor.
- A stable, crystal-driven 40Khz clock to modulate IR and sonar.
- A 16 pin DIP socket for mounting resistors or other discrete components in series with the digital outputs.
- A 60-pin header providing direct mating to a connector added to the EVBU board.

Mobile Platform

A TJ platform, which is produced by Novasoft, is used for this project because of it's small shape and light weight. Those reasons will help the robot avoid obstacles easier and save energy. The platform was modified to fix the EVBU board. So, it still looks compact even with EVBU and ME11 boards on.

Actuation

1. Battery Power: two 6-AA NiCd battery packs (7.2 V). LM2930 regulated the voltage down to 5 V to supply to MCU and ICs.

2. Motor: Two Aristo MS410 servos are controlled by SN754410NE (1A max).

Sensors

The Mine Sweeper employs 4 types of sensors:

1) IR sensors (proximity and long-range)
2) Bump switches
3) Magnetic field sensor
4) Photodiodes (red color and flash light detection)

Three hacked Sharp GP1U58Y infrared sensors are used as the IR proximity detector and IR long-range detector. The IR proximity detector receives IR which reflects from objects in its path and IR long-range detector (beacon detector) looks for IR from the mines. The analog output is about 1.5 V - 2.5 V. The processor A/D converter provides digital outputs in the range 86 to 131 (about 5.5 bits of precision.)

Bump switches detect anything else that IR proximity detector can not see, or in case the robot backs up into the wall. The digital outputs are 0 (normally open) and 255 (close when hitting a obstacle.)

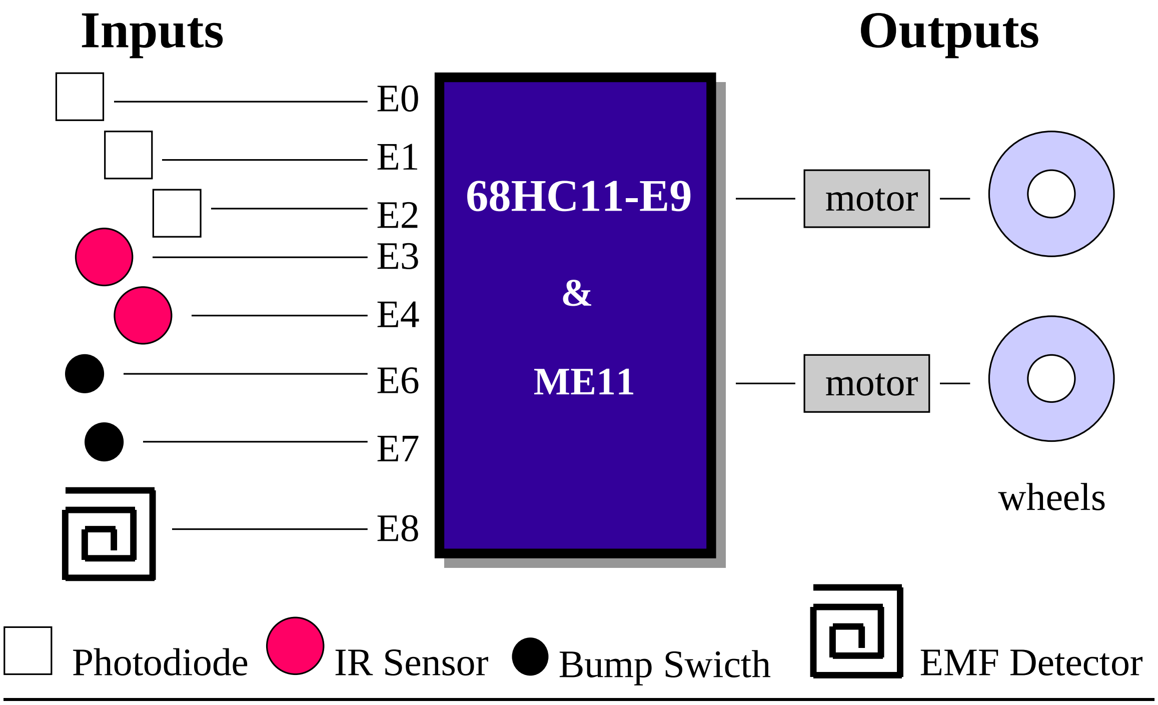
IMT magnetic field detector will help the robot avoid the magnetic mines in its path. Taking the advance of LED array to decide the threshold.

Burr-Brown OPT101 built-in amplifier photodiode is a little clear plastic 8-pin DIP. One photodiode is used as a red color detector to prevent from running into fixed mines. Another will be used as a flash light detector to go back to it starting location. The digital outputs is in the range 1 to 218 (about 7.5 bits of precision.)

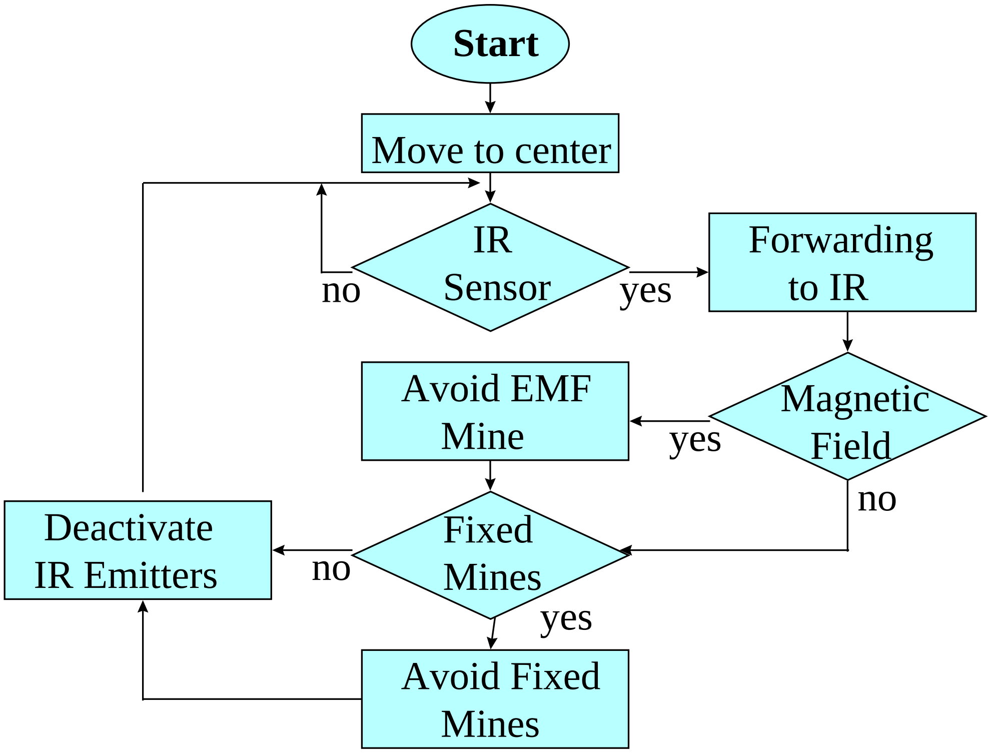
Behaviors

The Mine Sweeper knows to go straight, turn around (180^0 or 360^0) to look for IR or flash light, backup and go around when detect a fixed or magnetic mine. An IR emitter is assumed to be turned off when the robot detects the white area at the corner.

System Diagram

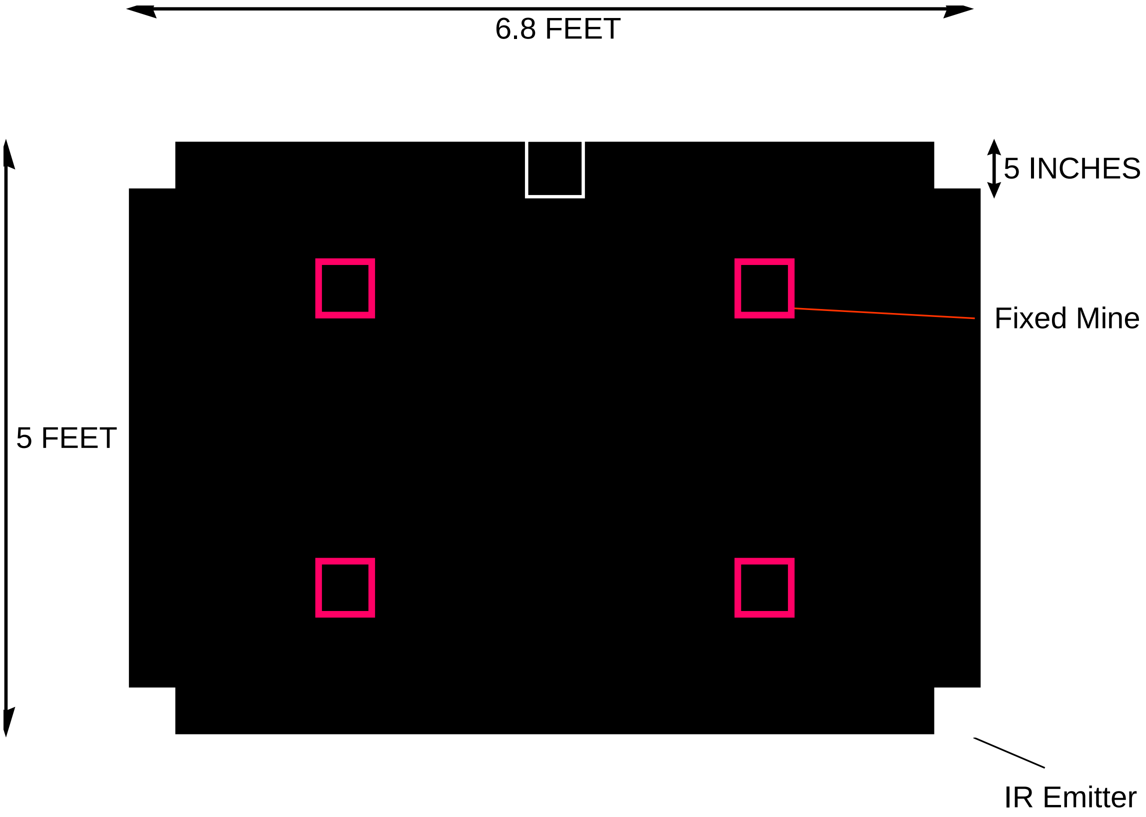


Program Structure



Experimental Layout

The robot will start from the middle of one of the sides of the mine field. It has to detect and turn off all 4 emitters at 4 corners. Only one corner will be turned on at a time. After completing the mission, the robot will look for a flash light signal and go back to its starting location. Besides the IR mines, 5 other mines are also on the field. Four of them are at fixed locations, and the magnetic mine is at random location.



Playing Surface

Fig. 1: Mine Field Lay Out

Cost Estimate

1.	Servo Kit: (MIL lab)	$40.00
2.	Sensor Kit: (MIL lab)	$25.00
3.	Novasoft ME11 Kit: (MIL lab)	$53.00
4.	Motorola EVBU M68HC11 - E9 board (68.00 + tax)	$72.20
5.	3 Photodiodes (estimated)	$12.00
6.	Magnetic sensor (estimated)	$10.00
7.	Miscellany (estimated)	$37.80
8.	Total (estimated)	$**250.00**

Conclusion

Mine sweeper is supported to perform object avoidance behavior, detecting IR, turning off IR emitter, and finding the way back to its starting location. Replacing the traditional CdS sensor with Photodiode (built-in amplifier) improves the red and white color detection rate and flashing light. The two little light bulbs that required 1.2V and 220mA each drain lots of current. So, I have to use another battery pack to prevent some crazy behaviors.

Documentation

F.G. Martin, The 6.270 Robots Builder’s Guide, 1992.

Motorola, MC68HC11 EVBU User’s Manual, 1992.

Assembly Manual, Mekatronix ME11 Expansion Board for the MC68HC11 EVBU, 1997

Keith L. Doty, Talrik Junior Assembly Manual, 1996

APPENDIXES

```
/*  Rudimentary obstacle avoidance using 2 front mounted IR sensors.
    and when detecting a red line. Find IR at corners
    By Nick Nguyen
    6 Oct 1997 */

/*--------------------------------------------------*/

void stop_go()
{
  motor(RIGHT_MOTOR,STOP);
  motor(LEFT_MOTOR,STOP);
}

void go_forward()
{
  motor(RIGHT_MOTOR,NORM_SPEED);
  motor(LEFT_MOTOR,NORM_SPEED-15.5);
}

void go_forward_right()
{
  motor(LEFT_MOTOR,NORM_SPEED);
  motor(RIGHT_MOTOR,STOP);
}

void go_forward_left()
{
  motor(RIGHT_MOTOR,NORM_SPEED-15.0);
  motor(LEFT_MOTOR,STOP);
}

/*--------------------------------------------------*/
void go_fast()
{
  motor(RIGHT_MOTOR,FAST_SPEED);
  motor(LEFT_MOTOR,FAST_SPEED-20.0);
}

/*--------------------------------------------------*/
void go_back()
{
  motor(RIGHT_MOTOR,-1.0*SLOW_SPEED/2.0);
  motor(LEFT_MOTOR,(-1.0*SLOW_SPEED/2.0)-4.0);
}

void go_back_left()
{
  motor(LEFT_MOTOR,-1.0*SLOW_SPEED);
  motor(RIGHT_MOTOR,STOP);
}

void go_back_right()
{
  motor(RIGHT_MOTOR,-1.0*SLOW_SPEED);
  motor(LEFT_MOTOR,STOP);
```

```
}
/*---------------------------------------------------*/
void go_around_right()
{
  motor(RIGHT_MOTOR,(-1.0*SLOW_SPEED)+5.0);
  motor(LEFT_MOTOR,SLOW_SPEED-4.0);
}

void go_around_left()
{
  motor(RIGHT_MOTOR,SLOW_SPEED-5.0);
  motor(LEFT_MOTOR,(-1.0*SLOW_SPEED)+5.0);
}

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

/*---------------------------------------------------*/
void read_sensors()
{
  while(1)
  {
    IR_LEFT=analog(6);
    IR_RIGHT=analog(7);
    TOUCH=peek(0x1000)&0x03;
    LEFT=analog(1);
    RIGHT=analog(0);
    CENTER=analog(2);
    MAGNET=analog(4);
    BEACON=analog(3);
    LIGHT=analog(5);
  defer();
  }
}
/*---------------------------------------------------*/
void avoid_obstacles()
{
  while (1)

   {
    if ((IR_LEFT > THRESHOLD2 && IR_RIGHT <= THRESHOLD1) || (LEFT > 65 && RIGHT < 55))
     {go_back_right();
      sleep(.2);}

    else if ((IR_RIGHT > THRESHOLD1 && IR_LEFT <= THRESHOLD2) || (LEFT < 50 && RIGHT >
50))
     {go_back_left();
      sleep(.2);}
    else if ((IR_LEFT > THRESHOLD2 && IR_RIGHT > THRESHOLD1) || (MAGNET > 40))
      {go_around_left();
       sleep(.3);}

    else if (TOUCH == FRONT)
      {go_back_left();
       sleep(.25);
       go_forward();}

    else if (TOUCH == REAR)
      {go_fast();
       sleep(1.5);}

/*     else if (CENTER > THRESHOLD3)
       stop_go(); */

    else go_forward();

    defer();
   }
}
/*----------------------------------------------------*/
void find_IR()
{
 while (1)
 {
  if (start==0)
    {go_forward();
     sleep(3.5);
     stop_go();
     start=1;}

  if (round < 4)
   {
   if (IR_LEFT >= 89 && IR_RIGHT <= 89 && search_IR==1)
      go_around_left();

   else if (IR_LEFT <= 89 && IR_RIGHT >= 89 && search_IR==1)
      go_around_left();

   else if (BEACON >= 90)
     {go_forward();
      search_IR=0;
      wait(300);}

   else if (BEACON <= 87 && permit==0)
      search_IR=1;

   else if (CENTER > 60)
     {search_IR=0;
      permit=1;
      round++;
      go_around_left();
      sleep(3.5);
      go_forward();
      sleep(2.0);
      wait(400);
      permit=0;}

   else go_around_left();

   } /* if round */
   defer();
  }
}

/*-----------------------------------------------------*/
void find_lite()
{
 while (1)
 {
  if (round >= 4)
   {
   if (round == 4)
    {go_around_left();
     sleep(1.0);
     go_forward();
     wait(300);
     round++;}

   if (LIGHT > 40 && round > 4)
     go_forward();
   else if (LIGHT < 40 && round > 4)
    {go_around_left();
     wait(300);
     go_forward();
     sleep(1.5);}
   }
   defer();
  }
}

/*-----------------------------------------------------*/
int start=0;
int round=0;
int permit=0;
int search_IR=1;
int LEFT, RIGHT, CENTER, IR_LEFT, IR_RIGHT, BEACON, LIGHT, MAGNET, TOUCH;

int FRONT=1;
int REAR=2;
int THRESHOLD1=120;
int THRESHOLD2=125;
int THRESHOLD3=20;

int LEFT_MOTOR=0;
int RIGHT_MOTOR=1;
float SLOW_SPEED=20.0;
float NORM_SPEED=50.0;
float FAST_SPEED=100.0;
float STOP=0.0;

void main()
{
  poke(0x7000,0x03);
  start_process(read_sensors());
  start_process(find_IR());
  start_process(avoid_obstacles());
/*  start_process(find_lite()); */
}
```