Surveillance Robot

Department of EEE
Ahsanullah University Of Science And Technology

Group No: 07

Group Members:

1# Touhid Jahan Rhidoy (12.02.05.073)

2# Muhafiz Ali (12.02.05.061)

3# Ainul Islam (12.02.05.065)

4# Fahim Shahrear Ahmed (12.02.05.013)

Introduction:

A surveillance robot is that robot which can moves in a certain area to collect necessary data in both ways at manually controlled or at auto controlled. It can collect the data by means of transferring the audio signal or transferring the video signal to a certain range of distance using WIFI.

Basic Principles:

(1) Standalone Arduino Uno:

The Arduino Uno is a microcontroller board based on the ATmega 328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

Basic Parts Of Arduino Uno:

Microcontroller : ATmega 8 A (Arduino Bootloaded)

Operating Voltage: 5V
Input Voltage: 7-12 V (through a LM 7805 REGULATOR)
Digital Pins: 14
Analog Pins: 6
DC Current: 40ma (PER PIN)
Flash Memory: 32 kb

(2) Motor Driver:

Two IC (L293D) is used to control 4 DC motors.

(3) HC-SR 04 Sonar Sensors:

They can do both send and receive the ultrasonic signals.

(4) HC – 05 Bluetooth Module

(5) Bluetooth App

Apparatus:

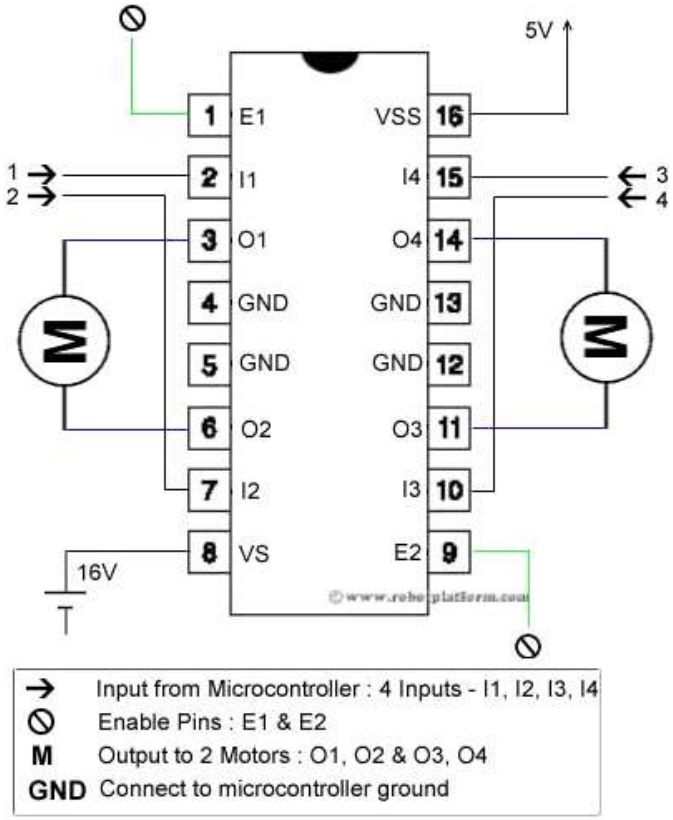
(1) Standalone Arduino
(2) Motor Driver (L293D)
(3) HC 05 Bluetooth MODULE
(4) Ultrasonic Sensor
(5) Robotic Chassis

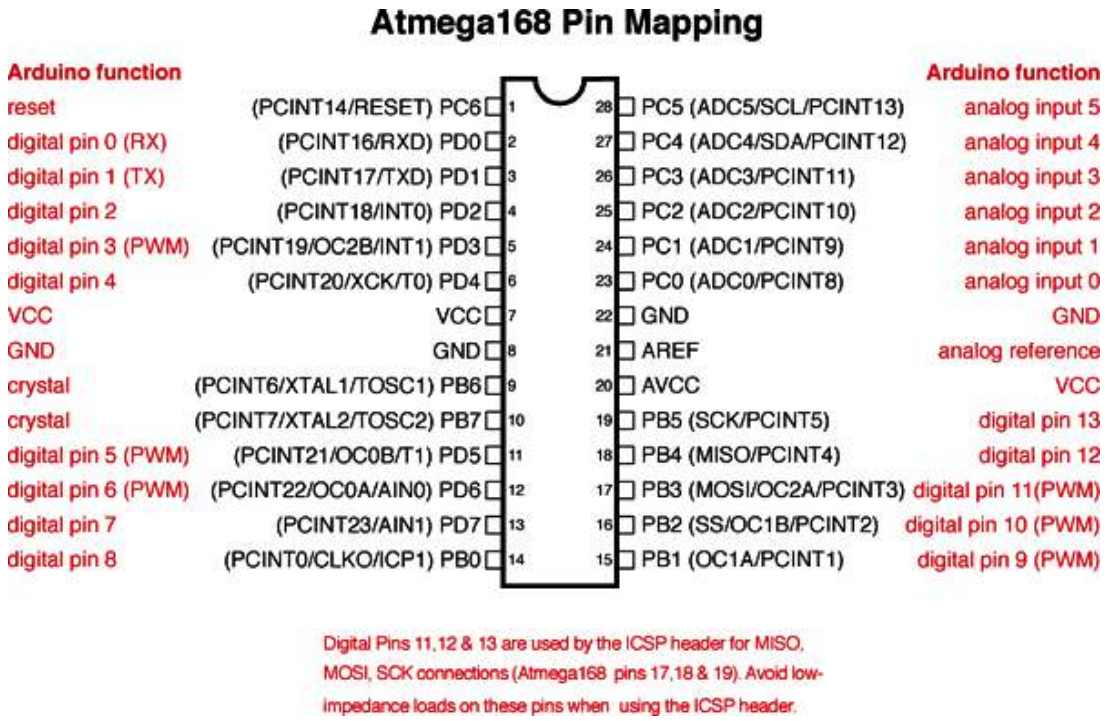
(6) DC Gear Motor
(7) Servo Motor
(8) Android Phone
(9) Batteries
(10) Others

CIRCUIT DIAGRAM

ATMEGA 8A ARDUINO COMPATIBLE PIN MAPPING (COMPATIBLE WITH ATMEGA 168 AND 328P)

L293D PINOUT





CONNECTION:

1st Motor Pin 1= Digital pin 2,

1st Motor Pin 2= Digital pin 3,

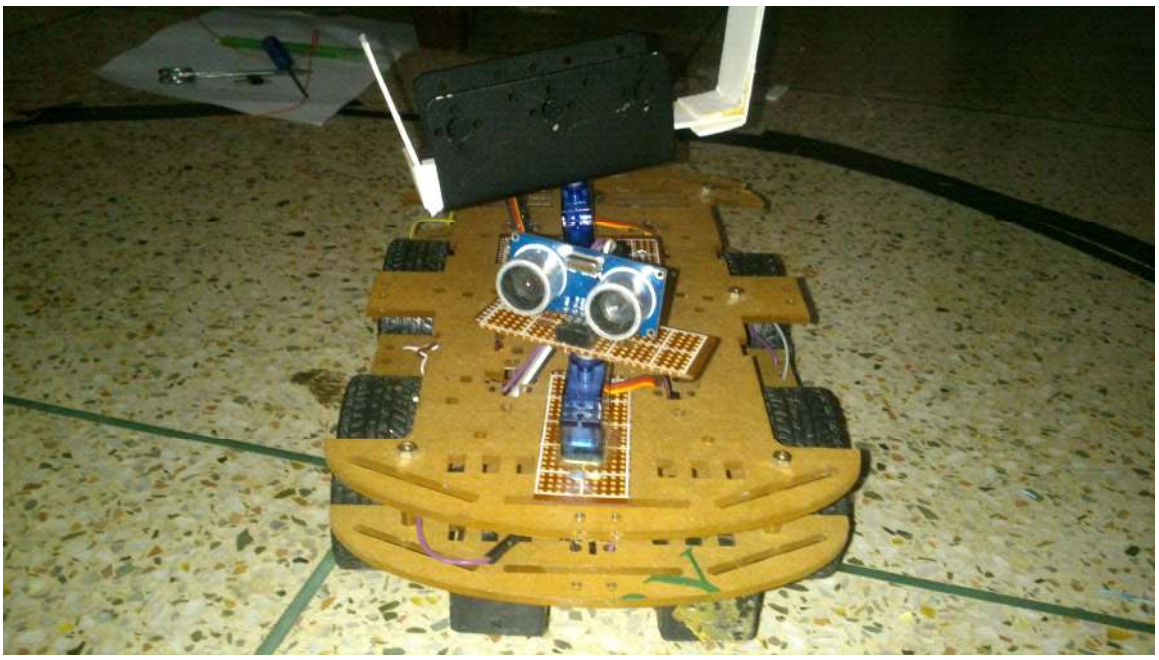
2nd Motor Pin 1= Digital pin 11,

2nd Motor Pin 2= Digital pin 12,

Servo 1= Digital pin 10,

Servo 2= Digital pin 11,

Power supply : DC 9V



Code:

```
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define LEDPin 13 // Onboard LED

#include <Servo.h>
  int maximumRange = 200; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration, distance,d1=0,d2=0,u=0,d3=0; // Duration used to calculate distance

Servo servo1;    // create servo object to control a servo
                 // a maximum of eight servo objects can be created

int pos = 0;
```

```
char mode='V',control,direcTion;



void setup()
{
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
  servo1.attach(10);


digitalWrite(13,HIGH);
}

void loop()

{
  if (Serial.available() > 0)
  {
    char x;
    x= Serial.read();
    delay(50);

    if (x=='v'||x=='V')
    {
      mode=x;
    }

    if (x=='F'||x=='R')
    {
      direcTion=x;
```

```
    }
    if (x!='v' || x!='V') control=x;
  }

  if (mode=='V')
  bcontrol();
  if (mode=='v')
  scontrol();

}

void off()

{
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(11,LOW);
  digitalWrite(12,LOW);

}

void left()
{
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(11,HIGH);
  digitalWrite(12,LOW);
}


void right()
{
  digitalWrite(11,LOW);
  digitalWrite(12,LOW);
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
}
```

```
void forward()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(11,HIGH);
  digitalWrite(12,LOW);
}

void backward()
{
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(11,LOW);
  digitalWrite(12,HIGH);
}

void Rleft()
{
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(11,LOW);
  digitalWrite(12,HIGH);
}

void Rright()
{
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(11,LOW);
  digitalWrite(12,LOW);
}

long sonar()
{
  digitalWrite(trigPin, LOW);
 delayMicroseconds(2);

 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
```

```
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.
 distance = duration/58.2;
 return distance;
}

void scontrol()
{
   u=sonar();
delay(5);

if (u>30) forward();
else
{
   off();
   delay(50);

   servo1.write(0);
      delay(1000);
       d1= sonar();



      delay(100);

      servo1.write(180);
      delay(1000);
       d2= sonar();




      delay(100);
       servo1.write(90);
       d3= sonar();

      delay(1000);
```

```
      if (d2<30 && d1<30)
     {backward();
     delay(1000);}
     else if(d1>=d2)
    { right();delay(1000);}

    else if (d2>d1)
    {
      left();delay(1000);
    }

}
}

void bcontrol()
{

 if (control=='S')
 off();
  else if (control=='F')
  forward();
  else if (control=='B')
  backward();
  else if (control=='L')
  {
  left();}

  else if (control=='R')

  right();

  else if ( control=='7'||control=='7'||control=='8'||control=='9'||control=='10')
  servoright();
  else if (control=='1' || control=='2'||control=='3'||control=='4')
  servoleft();

}
```

```
void servoleft()

{
   for(int x=1;x>0;x--)

  { digitalWrite(9,HIGH);
delay(2);
digitalWrite(9,LOW);
delay(15);}
delay(150);
}

void servoright()
{
   { digitalWrite(9,HIGH);
delay(1);
digitalWrite(9,LOW);
delay(16);}
delay (150);
}
```

Troubleshooting:

We Faced some problem with obstacle detection. In Some cases our robot was just taking decision. A few changes in the code solved the problem.

REFERENCES:

1. Standalone Arduino

   http://arduino.cc/en/Main/Standalone

2. Motor driver

   http://www.robotplatform.com/howto/L293/motor_driver_1.html

3. Bluetooth Control

   http://www.instructables.com/id/Arduino-AND-Bluetooth-HC-05-Connecting-easily/

4. Application (Arduino Bluetooth RC Car)

   https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller&feature=search_result

5. HC-SR 04 Sonar Sensors

   http://arduinobasics.blogspot.com/2012/11/arduinobasics-hc-sr04-ultrasonic-sensor.html