INTRODUCTION

For many people it is a machine that imitates a human—like the androids in Star Wars, Terminator and Star Trek: The Next Generation. However much these robots capture our imagination, such robots still only inhabit Science Fiction. People still haven't been able to give a robot enough 'common sense' to reliably interact with a dynamic world.

The type of robots that you will encounter most frequently are robots that do work that is too dangerous, boring, onerous, or just plain nasty. Most of the robots in the world are of this type. They can be found in auto, medical, manufacturing and space industries. In fact, there are over a million of these type of robots working for us today.

A robot has these essential characteristics:

- Sensing First of all your robot would have to be able to sense its surroundings. It would do this in ways that are not unsimilar to the way that you sense your surroundings. Giving your robot sensors: light sensors (eyes), touch and pressure sensors (hands), chemical sensors (nose), hearing and sonar sensors (ears), and taste sensors (tongue) will give your robot awareness of its environment.
- Movement A robot needs to be able to move around its environment. Whether rolling on wheels, walking on legs or propelling by thrusters a robot needs to be able to move. To count as a robot either the whole robot moves, like the Sojourner or just parts of the robot moves, like the Canada Arm.
- Energy A robot needs to be able to power itself. A robot might be solar powered, electrically powered, battery powered. The way your robot gets its energy will depend on what your robot needs to do.
- Intelligence A robot needs some kind of "smarts." This is where programming enters the pictures. A programmer is the person who gives the

robot its 'smarts.' The robot will have to have some way to receive the program so that it knows what it is to do

The diploma project is related to designing a Smart- Wheeled robot for obstacle avoidance.

Obstacle avoidance is one of the most important aspects of mobile robotics. Without it robot movement would be very restrictive and fragile. Obstacle Avoiding is a task which is used for detecting the objects placed in the path of your robot or any vehicle. So, to protect robot from any physical damages. An obstacle avoiding robot is an intelligent device, which can automatically sense and overcome obstacles on its path.

1 ARDUINO

Arduino is a popular programmable board used to create projects. It consists of a simple hardware platform as well as a free source code editor which has a “one click compile or upload” feature. Hence it is designed in way that one can use it without necessarily being an expert programmer (Kushner 1987). Arduino offers an open-source electronic prototyping platform that is easy to use and flexible for both the software and hardware. Arduino is able to sense the environment through receiving input from several sensors. It is also able to control its surrounding through controlling motors, lights and other actuators. The Arduino programming language that is based on the wiring and the Arduino development environment that is based on the processing are used to program the microcontroller found on the board (Banzi, 2005). Due to its open-source environment, one is able to easily write and upload codes to the I/O board. It is also worth to note that Arduino can be run on Linux, Mac OSX and Windows as its environment is written in Java

1.1 History

Arduino was released in 2005 by students from the Interaction Design Institute Ivrea (IDII) as a modest tool for Mac OSX and Windows. Since then, Arduino has been able to initiate an international-do-it yourself revolution at the electronics industry. The open source microcontroller hardware has been designed in a way that it can easily interface with various sensors (registering user inputs) and driving the behaviors and responses of the external components such as speakers, motors, and LED (responding to the user inputs). The most important feature of Arduino is the ease of programmability hence users with little expertise are able to use it. This aspect has made Arduino one of the most popular tools of choice for designers and artists in creating interactive spaces and objects (Arduino Team).

1.2 Development

While discussing the development of Arduino, it is worth introducing a brief history of microcontrollers. A revolutionary leap in the computing industry was seen in the 1960s following the development of solid state computers (including the IBM 1401), that used transistors to process its operations and a magnetic core memory for its storage (instead of vacuum tubes), and these enabled an increase in the compactness of the computer hardware. In addition, Jack Kilby’s invention of integrated circuits in 1959 enabled circuits and transistors to be fused into tiny chips of semiconducting materials (like silicon) as well as further miniaturization of the computer component. The other crucial development made in the same decade was the high level computer programming languages, written in symbolic languages such as plain English, and this made computer codes somehow easy to learn and read than the earlier machine languages that consisted of letters and numbers only. This development enabled individuals with few years of expertise to carry out the basic operations on a computer. FORTRAN (for the scientific calculators) and COBOL (for business application) were the two main languages that were introduced in that period. The microprocessor was one of the greatest innovations in the history of the modern 9 computer in the 1970’s. Initially, the microprocessor miniaturized all the hardware components of CPU to fit into one, tiny, integrated circuit, popularly known as the microchip. The microchip became the major driving component of the microcontrollers including Arduino which is made up of a microchip, input/output hardware and memory storage hardware for sensors. The microprocessor, due to the small form factor, was incorporated into a surfeit of electronic devices ranging from personal computers to calculators and are still used up to date. More programming languages were also developed in the

1970s and 80s including C, C++ and Java for applications in science and business. (Massimo, 2005)

1.3 Evolution

Having looked at the evolution of microcontrollers, there have been recent incarnations of the microcontrollers that have been designed in a way to fulfill the needs of hobbyists and casual users who happen to have a limited technical knowledge. In other words, the microcontrollers have moved from the more complex requirements in the scientific, business or commercial fields. Before the invention of Arduino, the PIC microcontroller board that was introduced by general instruments in 1985 was one of the most used tools for the electronic enthusiasts. The reasons as to why the PIC microcontroller board was preferred were the speed and ease of its programming through simple languages including PBASIC. An additional reason was that it was able to store programs on a flash memory chip that enabled the instructions on the board to be reprogrammed or erased at will with an infinite number of possibilities. It also supported output devices such as LEDs and motors as well as input sensors. There are other popular boards for the hobbyists including BASIC Stamp and wiring which are some of the microcontroller boards that were designed for tangible media explorations and electronic art. The two boards share the advantages of ease of rapid prototyping and simplicity of programming. It was in 2005 when the Arduino team was formed in Italy and it consisted of Barragan Massimo, David Cuartielles, Gianluca Marino, Dave Mellis and Nicholas Zambetti. The main goal of this team was to develop an electronic prototyping platform that would simplify the wiring platform and make it accessible to the non-technical users especially in the creative fields. The Arduino, therefore, incorporated several characteristics including a programming environment that is based on the processing language that was conceived by Casey Reas and Ben Fry and other artists and designers. Arduino also incorporated the ability to program its board using a standard USB connection with a low price point

1.4 Past and Present

Within its first 2 years of existence, Arduino achieved rapid success where more than 50, 000 boards were sold. By 2009, Arduino had more than 13 different incarnations with each having a specialized application. For instance, Arduino Mini was a miniature to be used in small interactive objectives, Arduino BT was built with Bluetooth capabilities, and Arduino Lilypad for wearable technologies projects. Today, the Arduino microcontroller is a popular prototyping platform across the world and it is a good example of how software and 10 hardware technologies that were originally created for business, military or scientific applications have been repurposed so as to serve the needs of people developing projects in new media and arts and design. The following list includes the present and past Arduino boards: Arduino shields, Arduino USB, Arduino single-sided serial, Arduino serial, Arduino Mega, Lilypad Arduino, Arduino Fio, Arduino BT, Mini USB adapter and Arduino Mini. In the development of Arduino, the following silverware times have been developed: in 2005, a project was started to develop a device that would control the student-built interactive design projects which was cheaper compared to other prototyping systems that were available at the time. The founders of the project, David Cuartielles and Massimo Banzi, named the project Arduin of Ivrea. They then began producing boards at a small factory in Ivrea in the Northwestern Italy. In September 2006, they released Arduino Mini and later in October 2008, Arduino Duemilanove was developed and was earlier based on Atmel ATmega 168 and later, on ATmega328. Arduino Mega was then released in March 2009 and was based on Atmel ATmega 1280. More than 300,000 units or Arduino in May 2011 were in use across the world. Arduino Leonardo was later released in July 2012 and is based on Atmel SAM3X8E that has an ARM Cortex-M3 core. Arduino Micro was released in November 2012 and is based on Atmel ATMega32u4. Arduino Uno has been named to mark the new Arduino 1.0 where version 1 and Uno will be the reference model for the Arduino platform. For the purpose of comparing with the previous versions, Arduino Uno will be used in the project. Arduino Uno is a microcontroller board which is based on the ATmega328 that has 14 digital I/O pins. Among these pins, six of them may be used as PWM outputs, one as a 16 MHZ crystal oscillator, 6 as analogue input, one as a USB connection, one as an ICSP header, and one as a power jack and reset button. Everything needed to support the microcontroller is contained on the board like connecting to a computer with a USB cable and power it using an AC-to-DC battery or adapter.

Arduino Uno differs from other preceding boards due to its features which include ATmega8U2 that is programmed as a USB-to-serial. Arduino Uno may be powered either through the USB connection or using an external power supply. The selection of power source is automatic and the power pins include: VIN which is the input voltage on the Arduino board while using the external power source and this is as opposed to the 5 volts from regulated power source or USB connection. Voltage may be supplied through this pin, or while supplying voltage through the power jack, it may be accessed through this pin. The regulated power supply is 5V that is used to power microcontroller as well as other components on the board. It can be supplied by the USB or any other regulated 5V supply. A 3.3V supply is generated by the on-board regulator or it can come from VIN through the on-board regulator. The maximum current drawn is 50 mA. The GND ground pins- the ATmega328- have 32 KB including 0.5 KB that is used for the boot loader. It has 2 KB of the SRAM and a further 1 KB of EEPROM that can be written or read with the EEPRO library. The maximum width and length or the Uno PCB is 5.3 and 6.8cm respectively. The power jack and the USB connector extend beyond these dimensions. 11

There are four screw holes that allow the board to get attached to a case or surface. It should be noted that the distance between the digital pins 7 & 8 is 0.16cm.

\

2 Design overview

2.1 Requirements

1. Vehicle will travel forward until obstacle is sensed.

2. Once obstacle is sensed vehicle will stop and turn until a clear path is found.

3. Vehicle will travel under own power.

2.2 Dependencies

1. Tiva C Launchpad microcontroller

2. HC-SR04 Ultrasonic range sensing module

3. Arduino Motor Shield v3 [L298 bridge]

4. Amico 12V 1:43 Geared Motors and Wheels

5. Batteries

2.3 Theory of Operation

Once power is applied the microcontroller begins with a drive forward routine. It will alternate between a delay and sounding for walls. This is essential to reduce echoes from previous sounds which will erroneously register as a wall. Once a wall is found it will reverse until it it no longer senses a wall. This gives the robot maximum clearance for turns. It will begin to turn left or right, chosen randomly. The turn routine alternates between a delay and sounding for the wall. The distance required for detecting a wall is increased by 33% during turns to ensure the turn is able to complete. The robot begins driving forward again. This cycle repeat until power is removed.

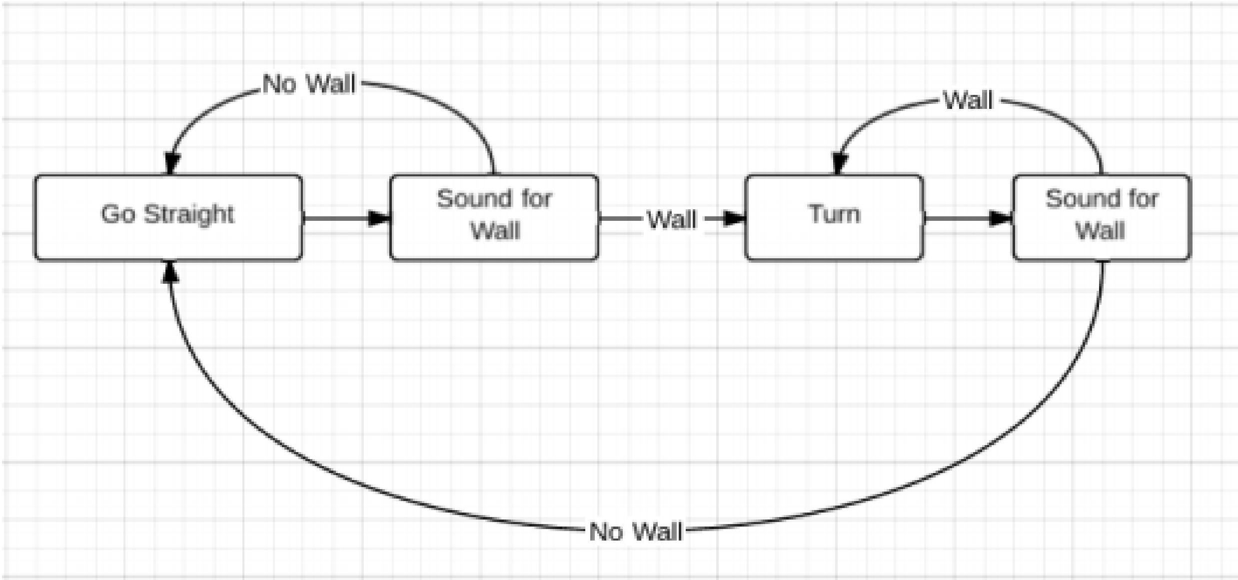

Figure 1- scheme of moving and turning of mobile robot

3 DESIGN DETAILES

3.1 Hardware

3.1.1 Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

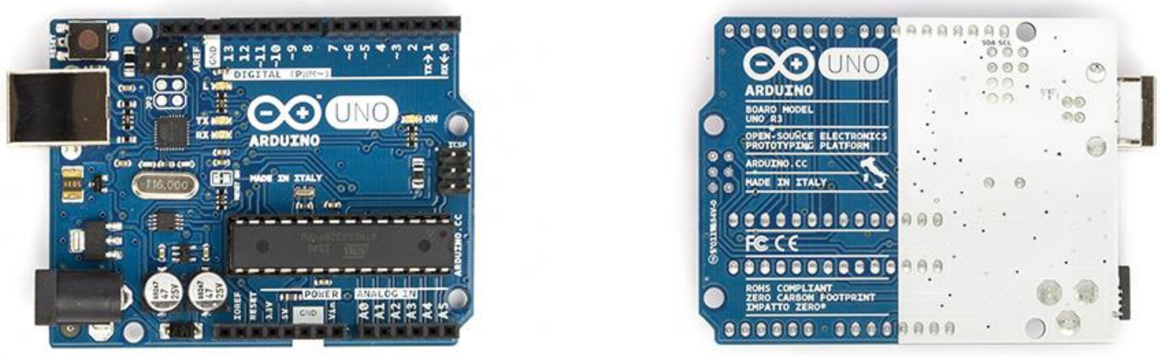

Figure 2- scheme of Arduino UNO

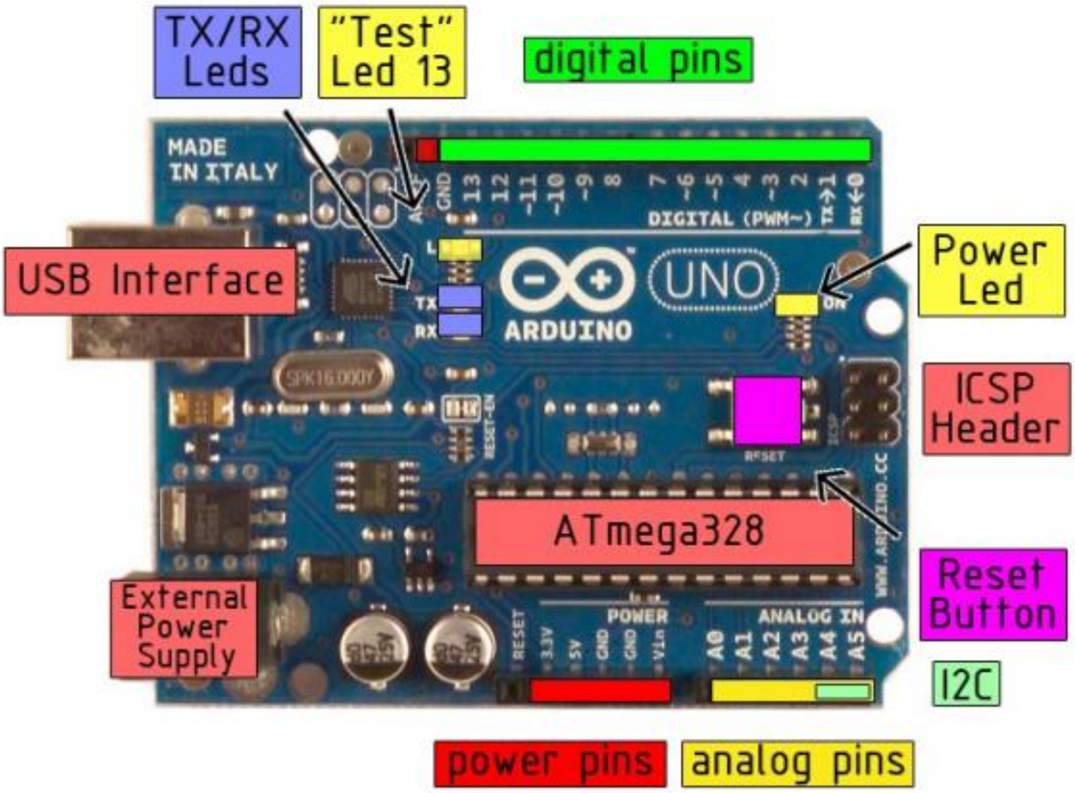

Figure 3- Pin allocation of Arduino Uno

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

• VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

• 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another
regulated 5V supply.

• 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

• GND. Ground pins.

3.1.2 L298P motor shield

This motor shield allows Arduino to drive two channel DC motors. It uses a L298N chip which deliveries output current up to 2A each channel. The speed control is achieved through conventional PWM which can be obtained from Arduino’s PWM output Pin 5 and 6. The enable/disable function of the motor control is signalled by Arduino Digital Pin 4 and 7.

The Motor shield can be powered directly from Arduino or from external power source. It is strongly encouraged to use external power supply to power the motor shield.

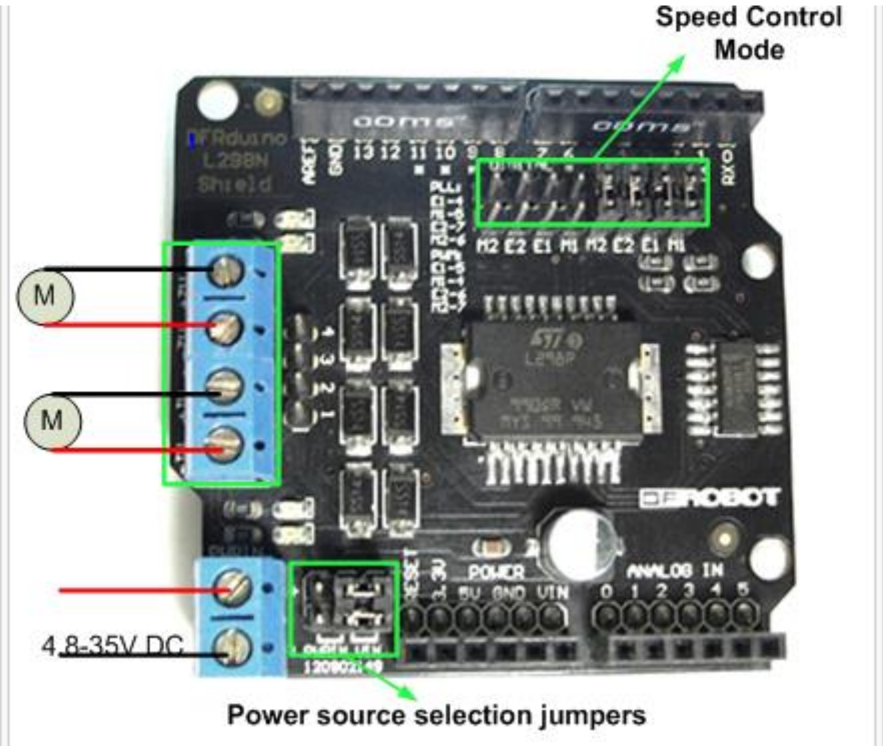

Figure 4- scheme of motor shield

- Logic Control Voltage：5V (From Arduino)
- Motor Driven Voltage：4.8～35V (From Arduino or External Power Source)
- Logic supply current Iss：≤36mA
- Motor Driven current Io：≤2A

- Maximum power consumption：25W（T=75℃）
- PWM、PLL Speed control mode
- Control signal level:

  High：2.3V≤Vin≤5V

  Low：-0.3V≤Vin≤1.5V

E1	M1		E2	M2	
L	X	Motor 1 Disabled	L	X	Motor 2 Disabled
H	H	Motor 1 Backward	H	H	Motor 2 Backward
PWM	X	PWM Speed control	PWM	X	PWM Speed control

Figure 5- scheme of controlling motor(dc motor)

"PWM Mode"

Pin	Function
Digital 4	Motor 2 Direction control
Digital 5	Motor 2 PWM control
Digital 6	Motor 1 PWM control
Digital 7	Motor 1 Direction control

"PLL Mode"

Pin	Function
Digital 4	Motor 2 Enable control
Digital 5	Motor 2 Direction control
Digital 6	Motor 1 Direction control
Digital 7	Motor 1 Enable control

Figure 6- scheme of pin allocation of motor shield

3.1.3 I/O expansion

This DFRobot ARduihno compatible I/O EXpansion Sheild is intelligently designed to facilitate an easy connection between an Arduino board (e.g. Arduino Duemilanove) and other devices such as sensors and RS485 devcices. In essence, it expands an Arduino controller's Digital I/O and Analogue Input Pins with Power and GND. It is compatible with Arduino Mega and is a perfect companion of Arduino Dumilanove (Atmega128 and Atmega 328).

Figure 7- scheme of I/O shield

This latest version (V5) IO expansion shield now supports Xbee. It combines our popular Xbee shield with our previous V4 IO expansion shield. Moreover, it supports SD card which provides the ultimate functional expansion for Arduino so far. As its predecessor, it still supports RS485, APC220, Bluetooth communication and servo control. It is compatible with Arduino Duemilanove (both ATmega 168 and ATmega 328) and Arduino Mega.

Features:

1. Supporting XBee (Xbee Pro)/Bluetooth Bee;

2. An unique RS485 output, supporting a RS485 device;

3. Separate PWM Pins, which are compatible with standard servo Connector;

4. Supporting Bluetooth module, APC220 module;

5. Auto Switch between external and onboard power supply;

6. Supporting SD card (read&write - our SD card module is needed);

Supporting IIC/I2C/TWI connection

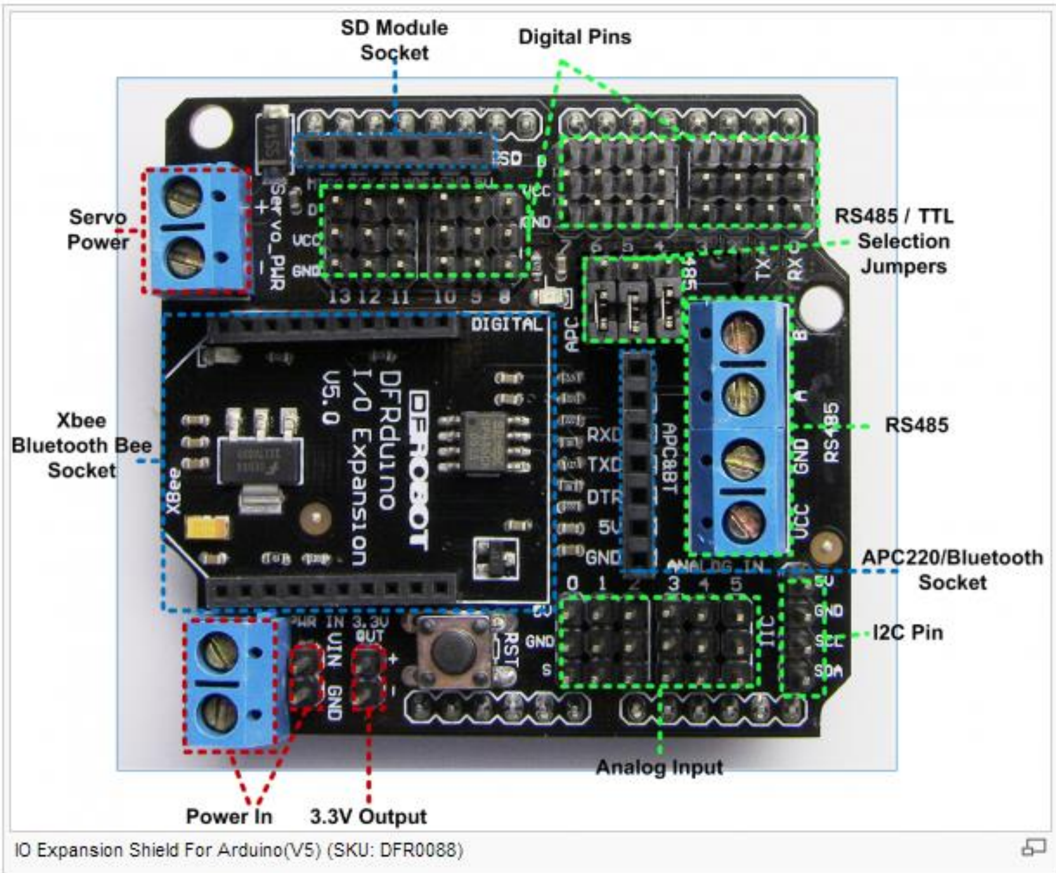

Figure 8- scheme of pin allocation of I/O shield

3.1.4 HC-SR04

The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ .

Figure 9- scheme of ultrasonic sensor

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1” to 13 feet. It operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

Features:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Currnt: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1" - 13ft

- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

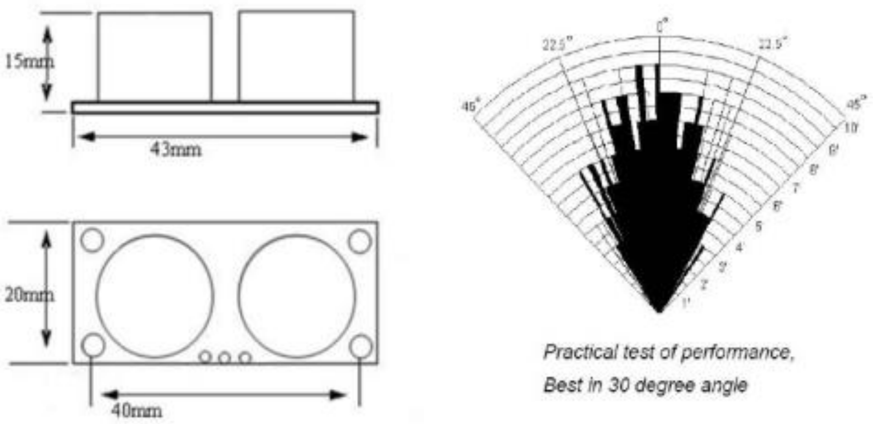

Figure 10- scheme of how sensor works

The timing diagram of HC-SR04 is shown. To start measurement, Trig of SR04 must receive a pulse of high (5V) for at least 10us, this will initiate the sensor will transmit out 8 cycle of ultrasonic burst at 40kHz and wait for the reflected ultrasonic burst. When the sensor detected ultrasonic from receiver, it will set the Echo pin to high (5V) and delay for a period (width) which proportion to distance. To obtain the distance, measure the width (Ton) of Echo pin.

Time = Width of Echo pulse, in uS (micro second)

- Distance in centimeters = Time / 58
- Distance in inches = Time / 148
- Or you can utilize the speed of sound, which is 340m/s

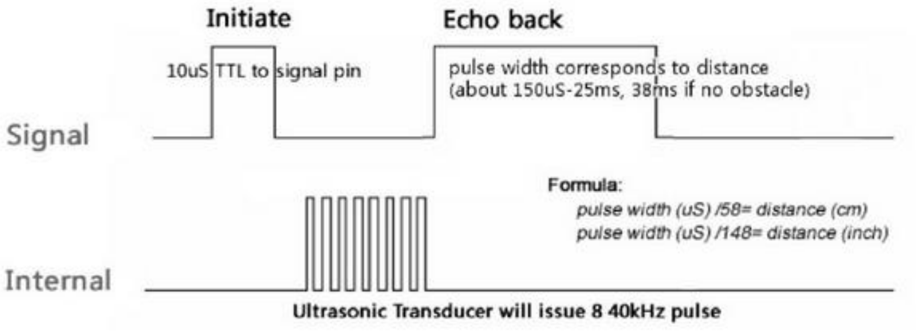

Figure 11- PWM of sensor

3.1.5 Power supplies

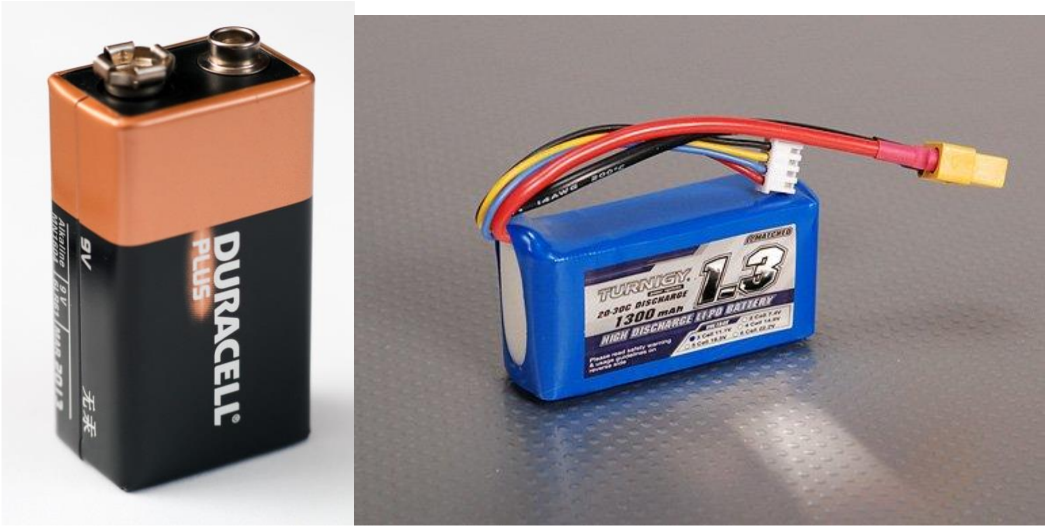

Figure 12- images of batteries

3.1.6 Switch and wires

Figure 13- images of wires and switch

3.1 Software

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

Open blink sketch

Sketches are little scripts that you can send to the Arduino to tell it how to act. Let's open up an Example Sketch. Go to the File menu -> Sketchbook -> Examples -> Digital -> Blink

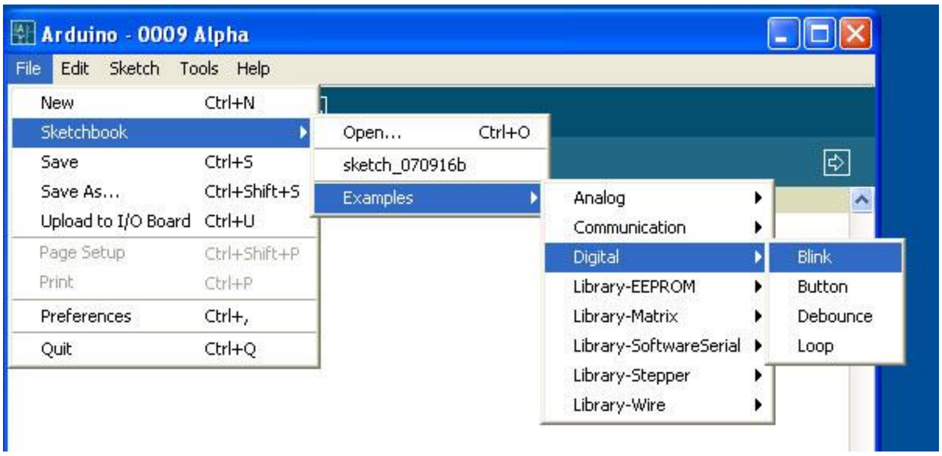

Figure 14- image of Arduino framework

The window should now look like this, with a bunch of text in the formerly empty white space and the tab Blink above it

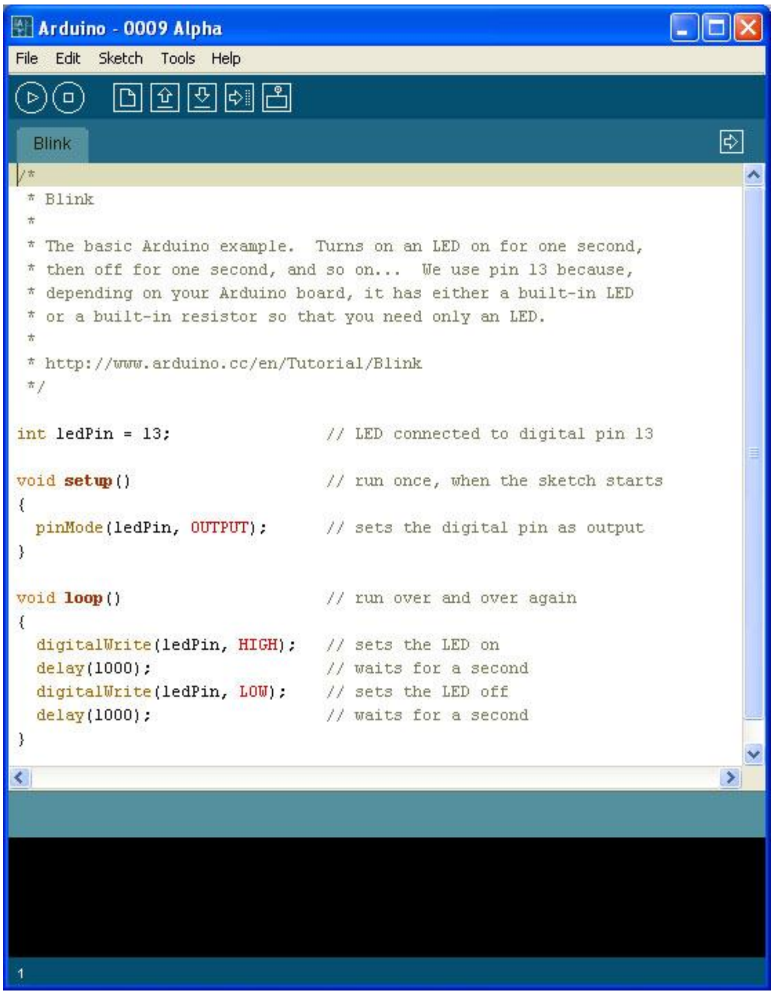

Figure 15- scheme of console of Arduino framework

Verify / Compile

The first step to getting a Sketch ready for transfer over to the arduino is to Verify/Compile it. That means check it over for mistakes (sort of like editing) and then translate it into an application that is compatible with the Arduino hardware.

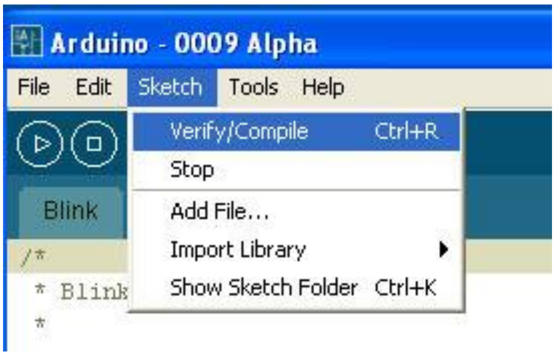

Figure 16- scheme that shows how to compile

After a few seconds, you should see the message Done compiling. in the Status Bar and Binary Sketch Size: in the Notification area. This means the sketch was well-written and is ready for uploading to the Arduino board!

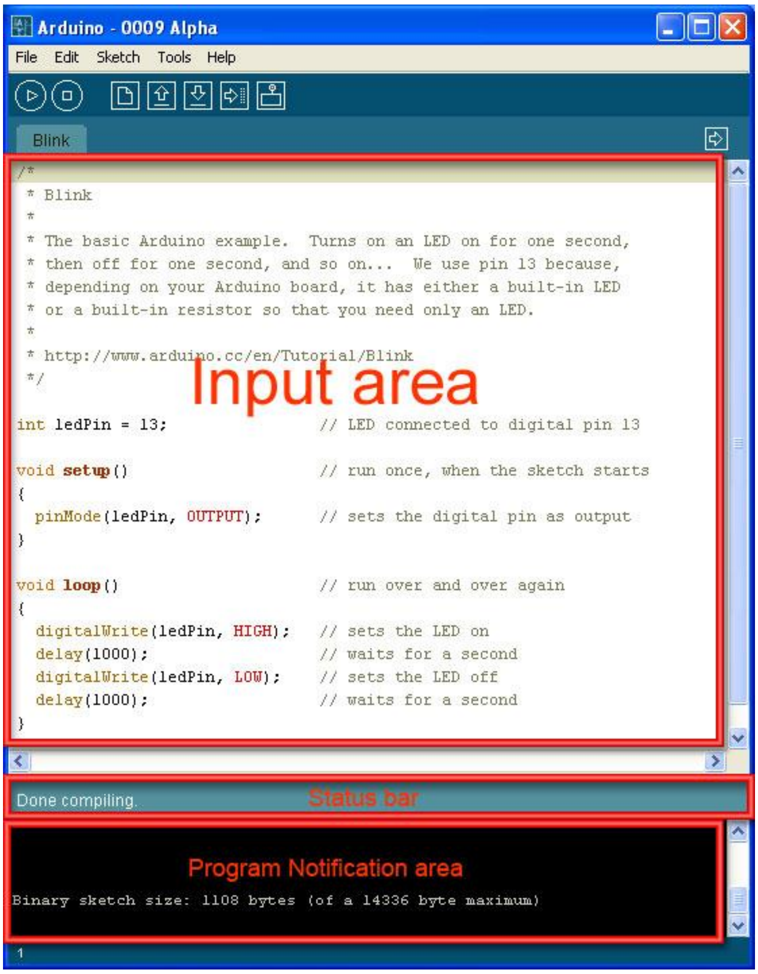

Figure 17- scheme that shows input area, status bar and program Notification area

Reset (NG only)

To tell the Arduino that it should prepare itself for a new Sketch upload, you must reset the board. Diecimila Arduino's have built-in auto-reset capability, so you don't need to do anything. Older Arduinos, such as NG, must be manually reset before uploading a sketch. To do that simply press the black button on the right hand side of the board, shown here.

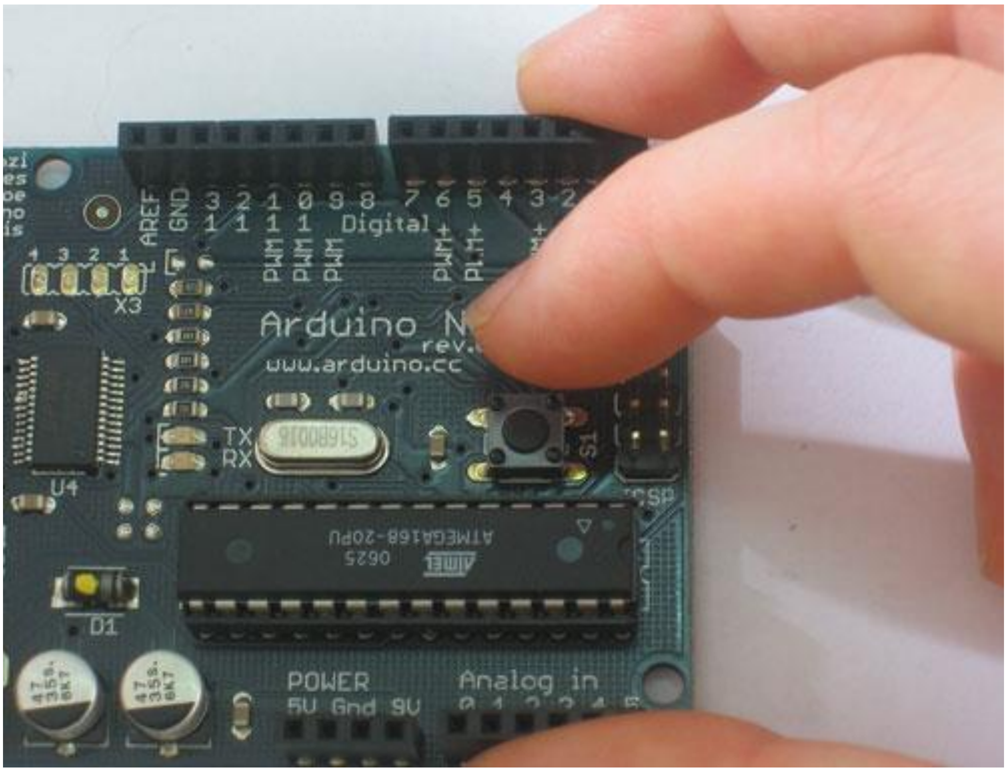

Figure 18- scheme that shows the allocation of reset button

Upload

Now it's time to upload. Make sure the Arduino is plugged in, the green light is on and the correct Serial Port is selected.

If you have an NG Arduino, press the Reset Button now, just before you select the Upload menu item.

Select Upload to I/O Board from the File menu

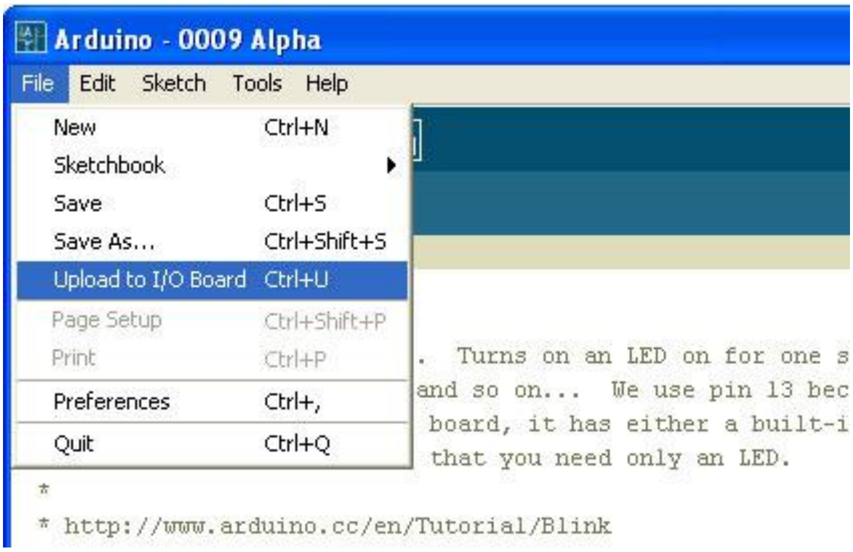

Figure 19- scheme that shows how to upload to contoller

After a few seconds you should get this screen, with the message Done uploading. in the status bar.

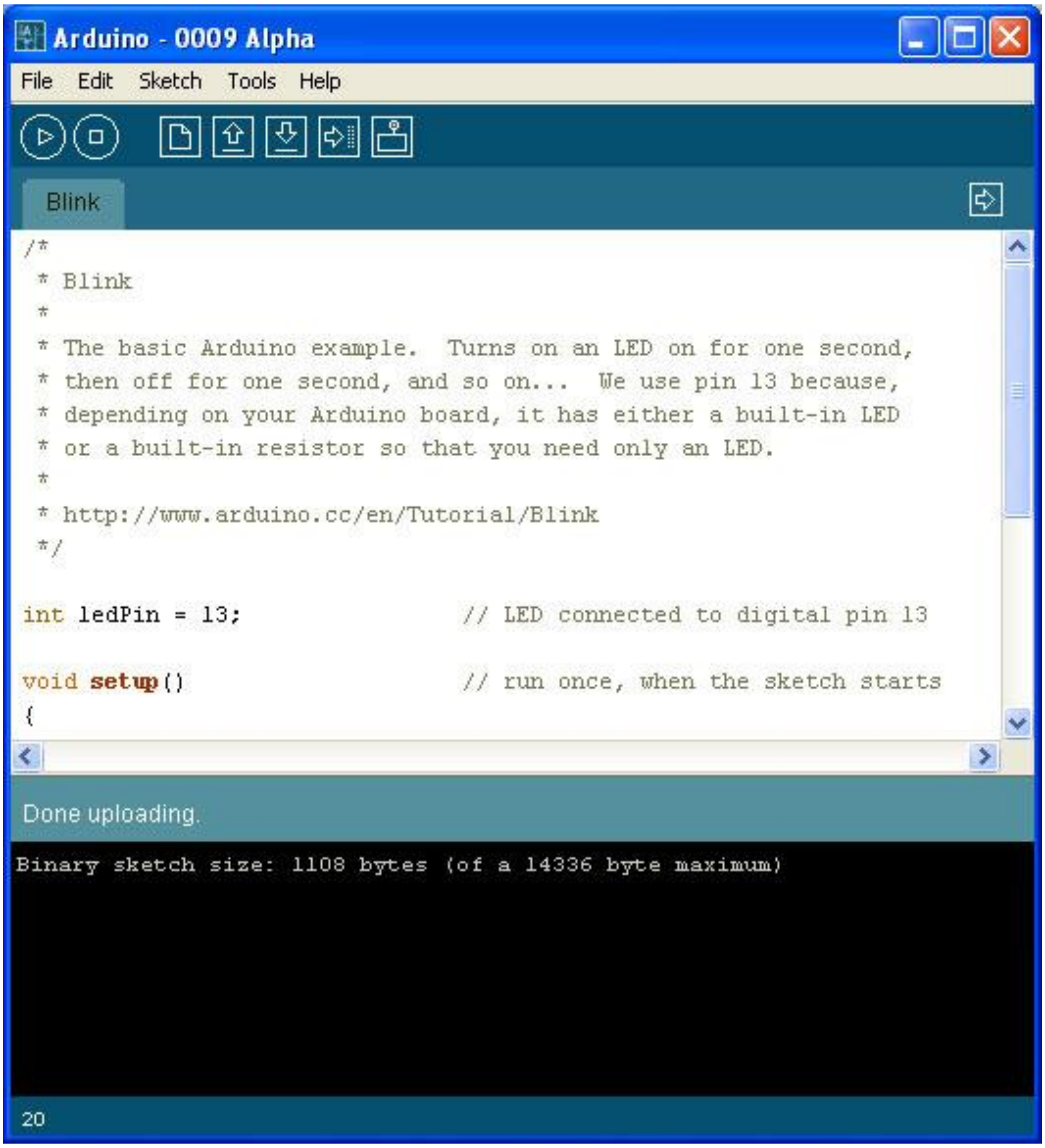

Figure 20- scheme that shows uploading was succesfull

If you get the following error message "avrdude: stk500_getsync(): not in sync: resp=0x00" that means that the Arduino is not responding

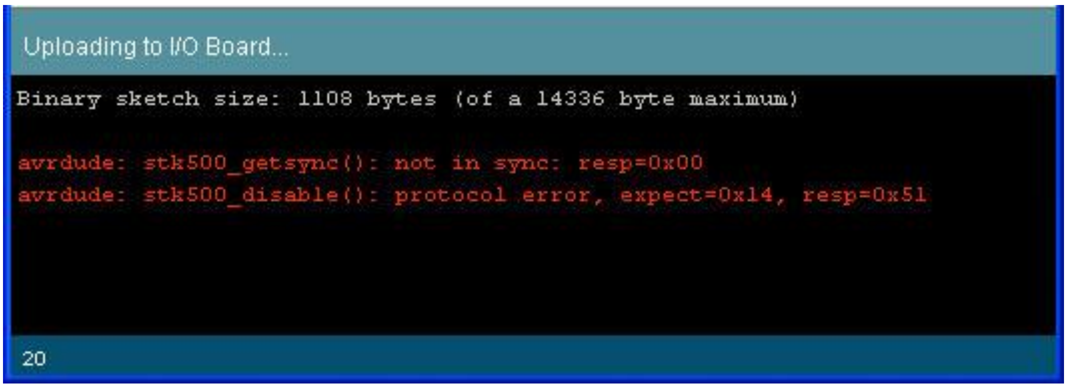

Figure 21- scheme that shows an error

Then check the following:

- If you have a NG Arduino, did you press reset just before selecting Upload menu item?
- Is the correct Serial Port selected?
- Is the correct driver installed?
- Is the chip inserted into the Arduino properly? (If you built your own arduino or have burned the bootloader on yourself)
- Does the chip have the correct bootloader on it? (If you built your own arduino or have burned the bootloader on yourself)

If you get the following error message:

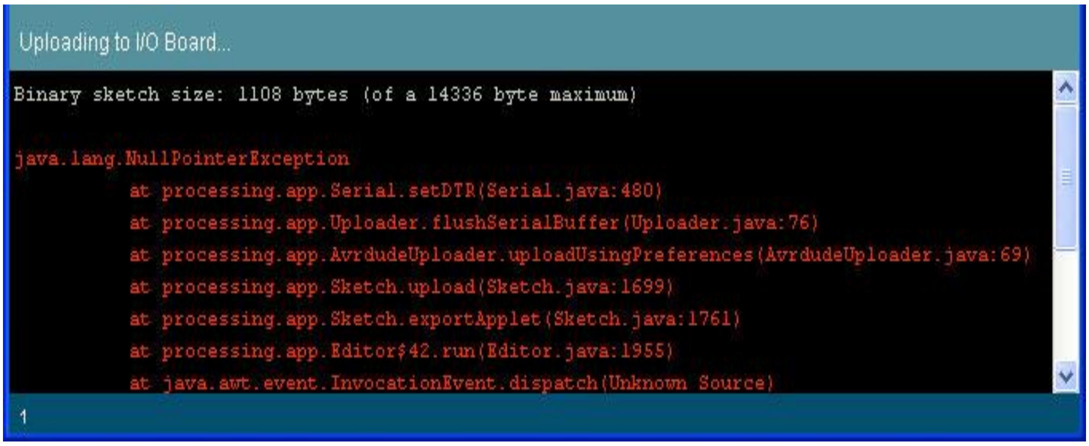

Figure 22- scheme that shows an error

It means you dont have a serial port selected, go back and verify that the correct driver is installed and that you have the correct serial port selected in the menu.

If you get the following error Expected signature for ATMEGA

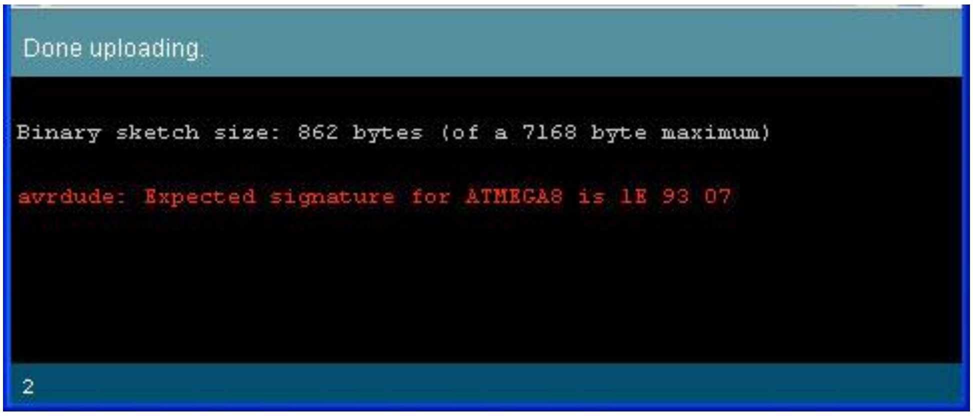

Figure 23- scheme that shows an error

Then you have either the incorrect chip selected in the Tools menu or the wrong boot loader burned onto the chip

If you get the following error: can't open device "COM10": The system cannot find the file specified (under Windows, COM port value may vary)

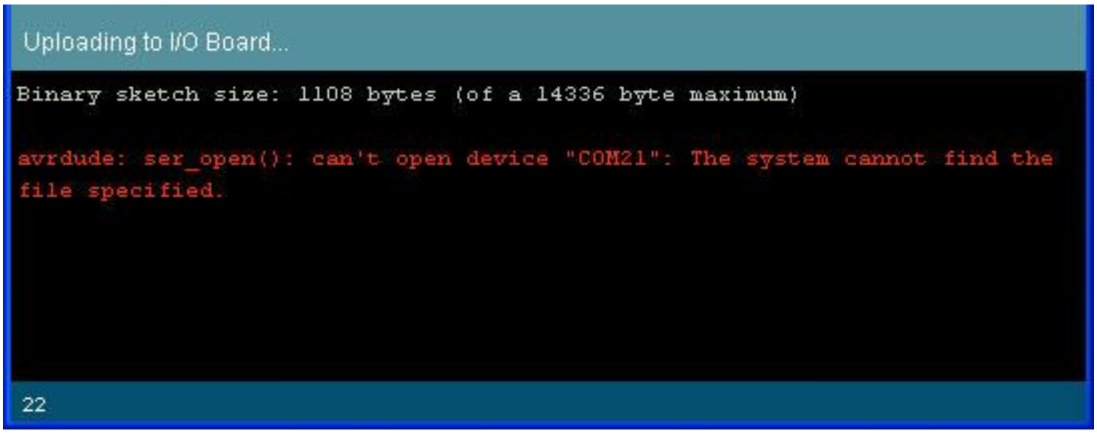

Figure 24- scheme that shows an error

It means that you have too many COM ports (maybe you've got 9 Arduinos?) You should make sure that the port is numbered as low as possible. You can use a program like FTClean to clear out old COM ports you aren't using anymore. Once you've cleaned out the ports, you'll have to reinstall the driver again

Alternately, if you're sure that the ports are not used for something else but are left over from other USB devices, you can simply change the COM port using theDevice Manager. Select the USB device in the Device Manager, right click and select Properties

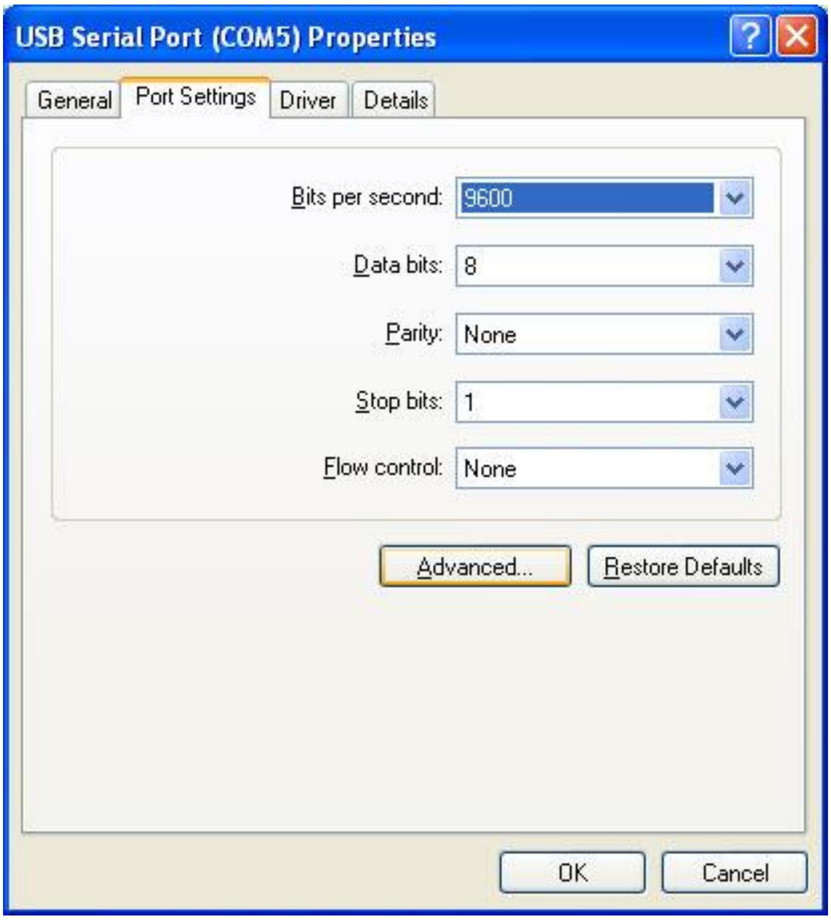

Figure 25- scheme of turning usb port components

Then click Advanced... and in the next window change the COM port to something like COM4 or COM5. Don't forget to select the new port name in the Arduino software. The lower port names may say (in use) but as long as the other USB devices aren't plugged in, it shouldn't be a problem. This is a little riskier than just using FTClean...

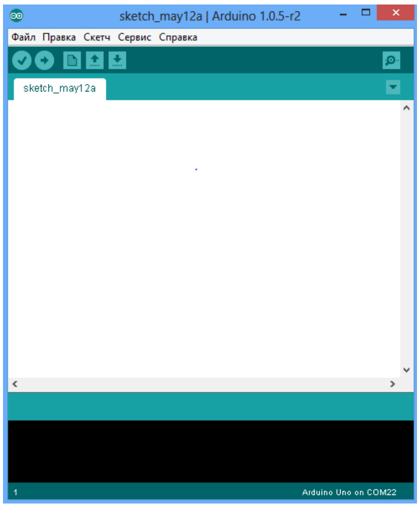

Figure 26- scheme of Arduino framework

In this example we’re using a single LED on the breadboard wired to the Arduino. As shown in the example, attach a ground wire (black) to the Gnd location on the bottom of the Arduino. Connect the wire to a ground rail on the breadboard. Using additional black wires jump to the bottom ground rail on the breadboard to complete a circuit adjacent to the LED position on the breadboard.

Place the LED in the breadboard and connect the black ground wire between the ground rail and the cathode lead (-) on the LED. Place a 150 to 220 ohm resistor in the breadboard. Attach another wire from one end of the resistor to the anode lead (+) of the LED. Finally, attach a wire between the remaining end of the resistor and Digital Pin 3 on the Arduino to compelete the circuit. This will allow us to write a program that controls flashing the LED.

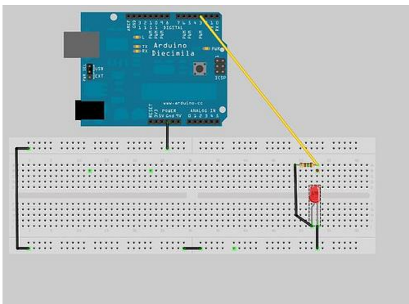

Figure 27- scheme of connection of led to Arduino Uno

4. TESTING

4.1 Rangefinder

Two sensors for measuring distance with the Arduino are extremely popular: the infrared proximity sensor and the ultrasonic range finder. They work in similar ways and achieve pretty much the same thing, but it's important to pick the right sensor for the environment you're in.

An infrared proximity sensor has a light source and a sensor. The light source bounces infrared light off objects and back to the sensor, and the time it takes the light to return is measured to indicate how far away an object is.

An ultrasonic range finder fires out high frequency sound waves and listens for an echo when they hit a solid surface. By measuring the time that it takes a signal to bounce back, the ultrasonic range finder can determine the distance travelled.

Infrared proximity sensors are not as accurate and have a much shorter range than ultrasonic range finders.

Consider the following during planning:

- Complexity: Both of these sensors are designed to be extremely easy to integrate with Arduino projects. In the real world, they're used for similar electronics applications, such as proximity meters on the back of cars that beep as you approach the curb. Again, the main complexity is housing them effectively.

  Infrared proximity sensors such as those made by Shape have useful screw holes on the outside of the body of the sensor. Maxbotix makes ultrasonic range finders that do not have these mounts, but their cylindrical shape makes them simple to mount in a surface by drilling a hole through.

- Cost: Infrared proximity sensors cost about $15 (£10) and have a range up to about 59 inches or less. Ultrasonic range finders have a far greater possible range and accuracy but an equally great price, costing between $27 (£18) for a sensor that can read up to 254 inches (645 cm) and $100 (£65) for a more weather-resistant model that can read up to 301 inches (765 cm).

- Where: A common application for these sensors is monitoring presence of a person or an object in a particular floor space, especially when a pressure pad would be too obvious or easy to avoid, or when a PIR sensor would measure too widely. Using a proximity sensor lets you know where someone is in a straight line from that sensor, making it a very useful tool.

IR proximity sensors are okay in dark environments but perform terribly in direct sunlight. The MaxBotix Ultrasonic Range Finder is one of the most reliable sensors. When using ultrasonic range finders, you can also choose how wide or narrow a beam you want. A large, teardrop-shaped sensor is perfect for detecting large objects moving in a general direction, whereas narrow beams are great for precision measurement.

In this example, you learn how to measure precise distances using a MaxBotix LV-EZ0. The EZ0, EZ1, EZ2, EZ3 and EZ4 all work the same way, but each has a slightly narrower beam, so choose the appropriate one for your project.

The range finder needs some minor assembly. To use the range finder in your circuit, you either need to solder on header pins to use it on a breadboard, or solder on lengths of wire.

You have three ways to connect your range finder: using analog, pulse width, or serial communication. In this example, you learn how to measure the pulse width and convert that to distance. The analog output can be read straight into your analog input pins but provide less accurate results than pulse width. This example does not cover serial communication.

You need:

- An Arduino Uno
- An LV-EZ0 Ultrasonic Range Finder

- Jump wires

Complete the circuit from the layout and circuit diagrams. The connections for the range finder are clearly marked on the underside of the PCB. The 5V and GND connections provide power for the sensor and should be connected to the 5V and GND supplies on your Arduino.

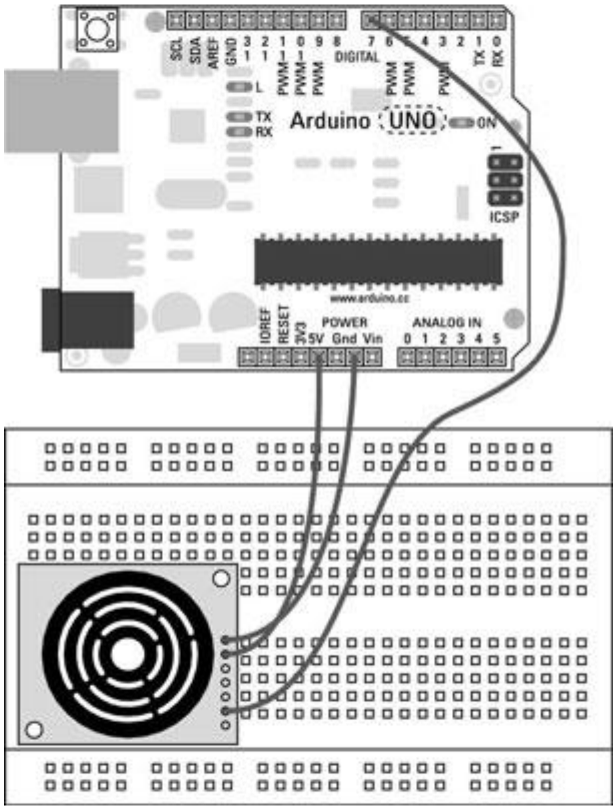

Figure 28- scheme of connection of sensor to Arduino Uno

The PW connection is the pulse width signal that will be read by pin 7 on your Arduino. Make sure that your distance sensor is affixed to some sort of base pointed in the direction that you want to measure.

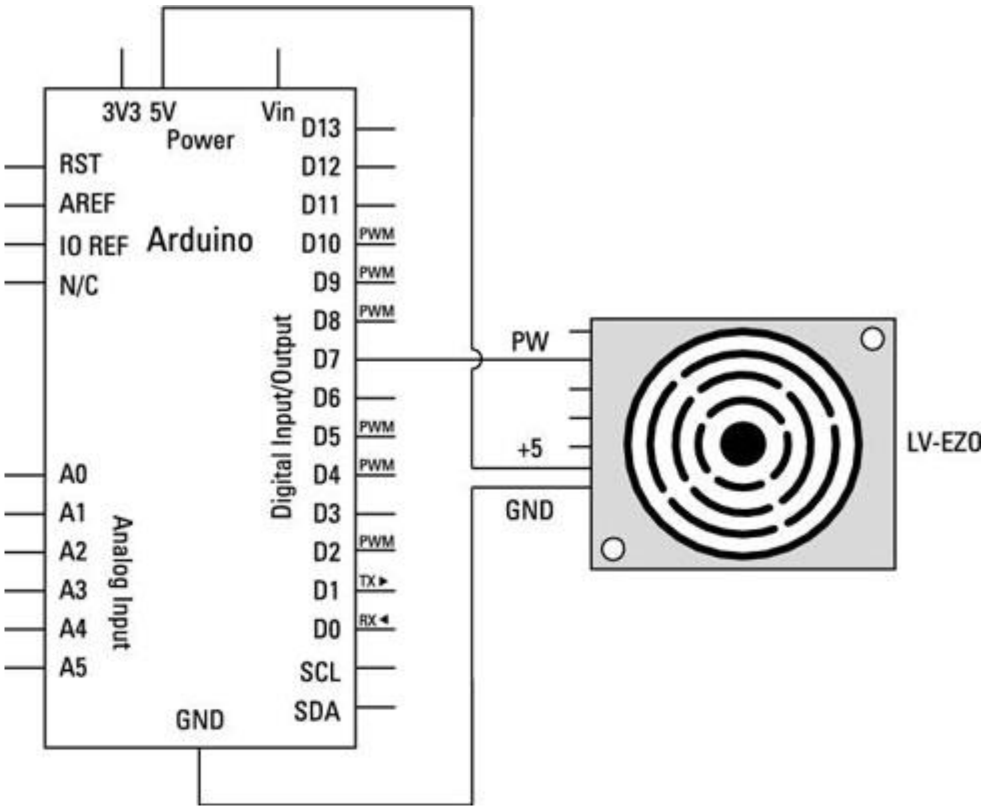

Figure 29- scheme of connection of sensor to Arduino Uno

4.2 Motor controller

To gain control of the speed of your motor whenever you need it, you need to add a potentiometer to your circuit.

You need:

- An Arduino Uno
- A breadboard
- A transistor
- A DC motor
- A diode
- A 10k ohm variable resistor
- A 2.2k ohm resistor

- Jump wires

Find a space on your breadboard to place your potentiometer. The central pin of the potentiometer is connected back to pin 9 using a jump wire, and the remaining two pins are connected to 5V on one side and GND on the other.

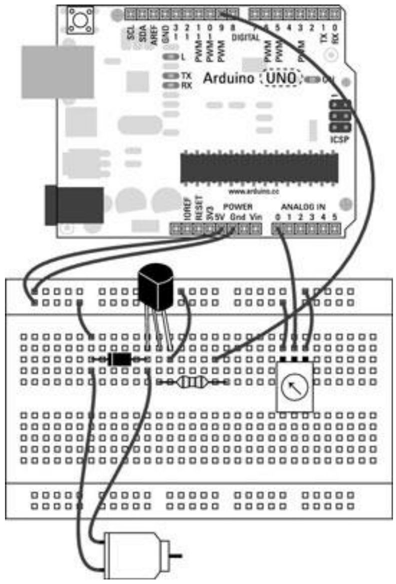

Figure 30- scheme of connection of DC motor to Arduino Uno

The 5V and GND can be on either side, but switching them will invert the value that the potentiometer sends to the Arduino. Although the potentiometer uses the same power and ground as the motor, note that they are separate circuits that both communicate through the Arduino.

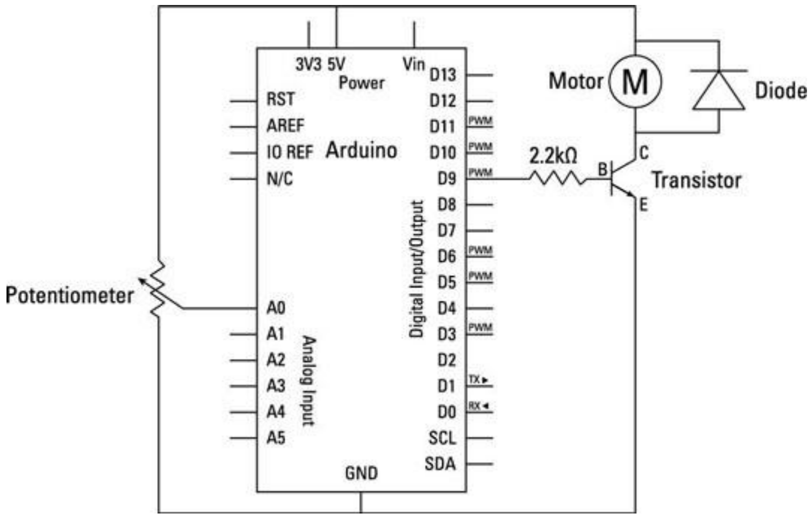

Figure 31- scheme of connection of DC Motor to Arduino Uno

After you have built the circuit, open a new Arduino sketch and save it with another memorable name, such as myMotorControl. Then type the following code.

```
int potPin = A0;
int motorPin = 9;
int potValue = 0;
int motorValue = 0;
void setup() {
 Serial.begin(9600);
}
void loop() {
 potValue = analogRead(potPin);
 motorValue = map(potValue, 0, 1023, 0, 255);
 analogWrite(motorPin, motorValue);
 Serial.print("potentiometer = " );
 Serial.print(potValue);
 Serial.print("\t motor = ");
 Serial.println(motorValue);
 delay(2);
```

}

After you’ve typed the sketch, save it and click the Compile button to highlight any syntax errors. .

If the sketch compiles correctly, click Upload to upload the sketch to your board. When it is done uploading, you should be able to control your motor using the potentiometer. Turning the potentiometer in one direction causes the motor to speed up; turning it the other way causes it to slow down.This sketch is a variation on the AnalogInOutSerial sketch and works in exactly the same way with a few name changes to better indicate what you are controlling and monitoring on the circuit.As always, you declare the different variables used in the sketch. You use the potPin to assign the potentiometer pin and motorPin to send a signal to the motor. The potValue variable is used to store the raw value of the potentiometer and the motorValue variable stores the converted value that you want to output to the transistor to switch the motor.

```
int potPin = A0;
int motorPin = 9;
int potValue = 0;
int motorValue = 0;
```

You may find that there is a minimum speed after which the motor will just hum. It does so because it doesn’t have enough power to spin. By monitoring the values sent to the motor using the MotorControl sketch, you can find the motor’s minimum value to turn and optimize the motorValue to turn the motor within its true range.

To find the range of motorValue, follow these steps:

1. With the MotorControl sketch uploaded, click the serial monitor button at the top right of your Arduino window.

   The serial monitor window will show you the potentiometer value followed by the output value that is being sent to the motor, in this fashion:

potentiometer = 1023 motor = 255

These values are displayed in a long list and update as you turn the potentiometer. If you don’t see the list scrolling down, make sure that the Autoscroll option is selected.

2. Starting with your potentiometer reading a value of 0, turn your potentiometer very slowly until the humming stops and the motor starts spinning.
3. Make a note of the value displayed at this point.
4. Use an if statement to tell the motor to change speed only if the value is greater than the minimum speed needed to spin the motor, as follows:

   (a). Find the part of your code that writes the motorValue to the motor:

   ```
   analogWrite(motorPin, motorValue);
   ```

   (b). Replace it with the following piece of code:

   ```
   if(motorValue > yourValue) {
    analogWrite(motorPin, motorValue);
   } else {
    digitalWrite(motorPin, LOW);
   }
   ```

5. Now replace yourValue with the number that you made a note of.

   If the value motorValue is greater than that, the motor speeds up. If it is lower than that, the pin is written LOW so that it is fully off. You could also type analogWrite(motorPin, 0) to accomplish the same thing. Tiny optimizations like this can help your project function smoothly, with no wasted movement or values.

5 KINEMATIC CONTROL OF THE MOBILE ROBOT AT MOVEMENT ON THE RANGE WITH OBSTACLE

5.1 KINEMATIC CONTROL

Now roughly the section of a robotics which is engaged in creation develops mobile robots. However to create the robots which are surely moving even on an equal surface on which there are obstacles insuperable to them, yet it wasn't possible on the reasons, including because of imperfection of algorithms of management. Therefore a development algorithm of management of mobile robots is an actual task.

This paper considers the problem of providing the desired motion of the mobilerobot by polygon with a stop at predetermined points and avoiding obstacles . A similar problem arises in the use of robots to deliver meals in hospitals , clearance of suspicious objects , fire , etc.

In the planning phase of the trajectory using the standard wave algorithm that allows to parameterize a route using cubic splines .Devel-oped law of kinematic control platform that provides movement formed by a curved path for a specified time . Introduction of compulsory equation relationship between linear and angular deviations enabled us to reduce the problem of synthesis to the elementary problem in which the number of controlled variables and control the same. The results ofnumerical simulations confirm the efficiency of the developed approach .

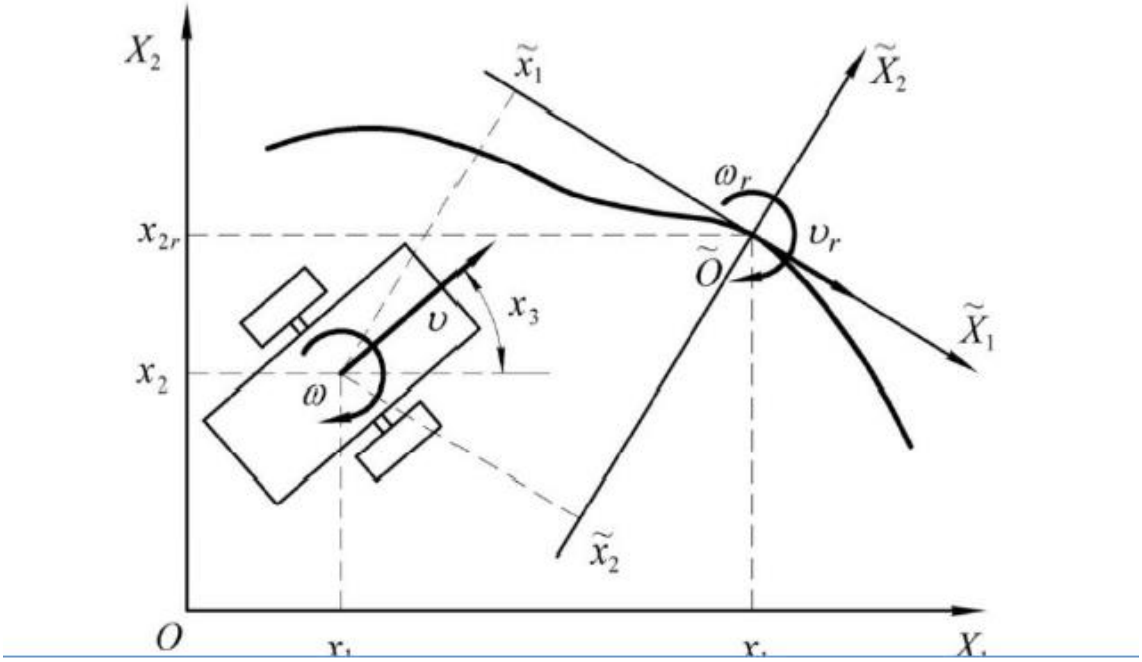

Figure 32- scheme of kinematic control of mobile robot

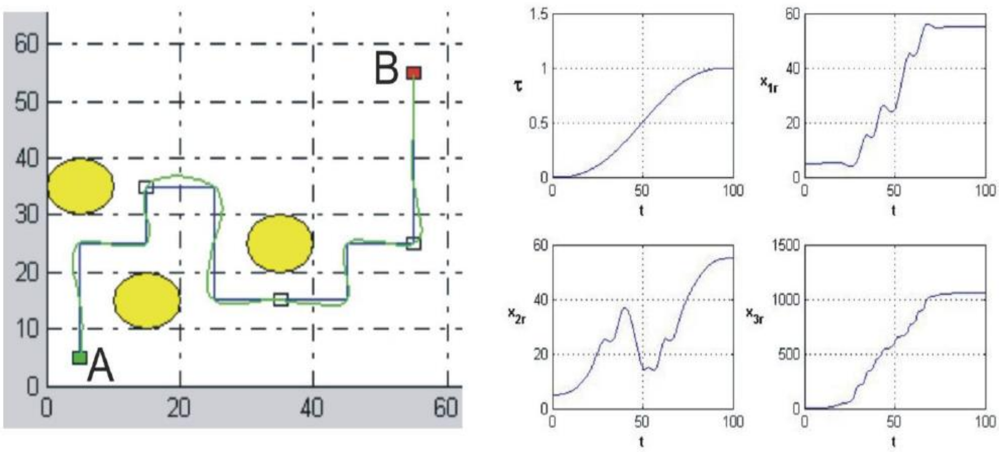

Figure 33- Robot trajectory(at the left) and basic variables(on the right)

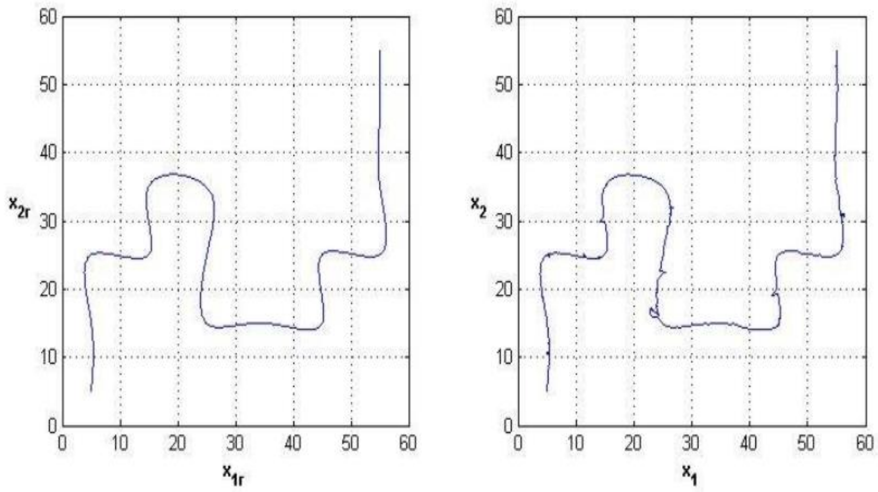

Figure 34- Basic trajectory(at the left) and results of modeling(on the right)

The offered approach and algorithm of formation of a program trajectory allow to operate robot movement effectively. Despite seeming complexity, algorithm of management it can be easily realized in the on-board computer of the mobile robot and it can be used as the main part of the general algorithm at control system creation. Thus offered algorithm of formation of a program trajectory can serve as basic at creation of the algorithms allowing the mobile robot to pass between obstacles of the district to go round group of obstacles, and also it can be puti n a basis of a flexible control system of robot movement.

6 The Economic part

6.1 The economic importance

Many States consider robots a key element of economic development, and invests millions of dollars in research. Manufacturing robots in the XXI century could become the largest industry sector:

a) Robots can be widely used for the extraction of raw materials and resources. Such robots can work in harsh and dangerous conditions such as radiation, climate and space.
b) Mining operations topical next 30-40 years. In the future, in connection with the arrival of nano-technology will be able to get the raw state of the scattered in the environment. Personnel crisis will hamper the development of deposits. Application of robotic technology will reduce the time to do the job and create the foundation for the country's economic prosperity in the future.
c) Most of the population is concentrated in the big cities. Now it is difficult to find people willing to work in remote areas of the country. Humanoid robots will play an important role in remote areas. There, they require significantly less costly to use than a person's work. However, a person's work also needed. After all, someone has to control robots. We need people who are engaged in service and additional training robots.
d) There are large companies that face the challenge of the future staff shortages in the production of various products. The robots will perform most of the work on the production, while people will be able to spend more effort and resources to develop more competitive products, and promotion of these products in the international markets.
e) Robots themselves are a product of high technology. Their development and implementation in production requires the development of an entire field of science and industry. The knowledge gained in the development of a variety of robots can be applied in various fields. Robots, components and parts,

software, all of this is a high-quality product that is sure to find a buyer in the near future.

The purpose of this calculation is to show the profitability of the domestic robot development.

6.2 Calculation of economic efficiency from the implementation of development

During the development of the technical design products, after theoretical calculations and constructive study of manufactured and configured prototype. All costs associated with the development, manufacture and customization prototype products listed in this section. Value prices for basic materials, purchased products and tariff rates for workers taken on April 1, 2014.

Manufacturing conditions: finished product, located in the housing development of the device in terms of production, produced - 20 pcs.

Product development costs calculated according to the formula:

$$Cost = C_{b.m.} + C_{p.p.} + WF + C_{s.t.} + C_{m.o.} + C_{over}$$

$C_{b.m}$ - The cost of basic materials;

$C_{p.p}$ - Costs of purchased products;

WF - wage fund

$C_{s.t.}$ - Social tax;

$C_{m.o}$ - Expenses for maintenance and operation of equipment

C_{over} - Overhead.

6.3 Costs of basic materials

Costs of basic materials in the development of robot board, may generally be calculated by the formula:

$$C_{b.m1} = \sum (m_i \cdot C_i),$$

where m_i - material consumption rate products, kg/ pcs; C_i - the price of one kilogram of material, tenge.

Table 1- Costs of basic materials

Name of purchased products	Type	Number of pieces	1 pc price, tenge	Amount per product tenge
Microcontroller	Arduino UNO	1	5000	5000
Motor controller	L298P	1	1500	1500
I/O shield	DFROBOT I/O Expansion	1	2000	2000
Ultra sensor	HC-SR04	1	2000	2000
Switches	Switches	1	900	900
Wires	Arduino wires	10	60	600
Battery(9V)	VARTA	1	900	900
Battery(11.1V)	Ardrone	1	3000	3000
TOTAL				15900

When calculating the cost of the product, please note that the cost of finished products supplier $C_{p.p}$ was charged VAT at 12%, so the cost of finished products in pure form, i.e. VAT will be

$$C_{b.m} = C_{b.m1} /1.12$$

$$C_{b.m}= 15900/1.12 = 14196.428 \text{tenge.}$$

6.4 Costs of metalwork, assembly and software development

To determine the basic and additional salaries for metalwork, assembly and software development, to first calculate the amount of wages on product development

Size of the basic wage of production workers CO serving main production per product is 15000 tenge. Development for microcontrollers for a product is 40,000 tenge and development for client-side is 60,000 tenge.

Payroll workers determined by the formula

$$WF = C_O + C_f$$

where C_O - basic salary, tenge; C_f - social tax tenge.

Costs of paying social tax MF constitute 11% of payroll wages fund

$$C_f = 115000 * 0.11 = 12650$$

Thus, payroll workers determined

$$WF = 115000 + 12650 = 127000$$

6.5 The cost of maintenance and operation of equipment

Next article - this is the article for the maintenance and operation of equipment, including a variety of costs associated with the maintenance and operation of workshop equipment, including:

- Depreciation of its value and the cost of repairs (cost of parts, salaries accruals repair workers, payment of maintenance of farms);

- Basic salaries, extra, extra-budgetary funds deductions subsidiary working workshop (i.e., electricians, engineers, toolmakers, transport workers in the workshop);

Depreciation and repair costs workshop vehicles wear low-value tools, accessories and auxiliary materials used in the normal course of the process , the cost of water is drawn from the water management system for production purposes.

Determining the costs of this article, based on a unit of production direct method is difficult or not possible. Here apply the method of calculating the estimated cost , i.e. calculated estimates (planned , regulatory , actual) as the sum of the costs attributable to the costs of maintaining and operating the equipment for a certain period in the whole shop, on the resulting value of the " spreads " by made during this period in the workshop production. In this case, using various methods of classifying these costs for specific products, mostly indirect methods, i.e., counting the cost estimate, is distributed among the products in proportion to the selected base. Often this basic salary of basic workers because its calculation is directly given process operations and their complexity.

Thus, the cost of maintenance and operation of the equipment can be calculated as follows:

$$C_{m.o.} = C_O \cdot K_{c.t}$$

where $K_{c.t}$ - coefficient taking into account the ratio of expenditure on maintenance and operation of equipment, (according to the commentary to GOST 2.601-2006 should choose $K_{c.t}$ ratio = 1.5 at the party less than 500 units, 1.0 at the party from 500 to 1000 units, 0.5 party with 1,000 or more).

$$C_{m.o.} = 115000 * 1,5 = 172500 \text{ tenge.}$$

6.6 Capital Costs

After calculating all cost items the capital costs, which are equal to the sum of all costs, i.e. overall development costs. Product development costs calculated by the formula above.

$$C = 15508.93+127000+172500 = 315008.93$$

When forming the price of the product is necessary to increase costs by 20%. Presented below is the formula for determining the price.

$$Cost = C + C \cdot 0.2,$$

$$Cost = 315008.93+ 315008.93 * 0,2 = 378010,716 \text{ tenge.}$$

Price subject to value added tax (VAT rate 12%), will be:

$$\text{Cost}_{\text{vat}} = 378010,716 * 1.12 = 423372$$

6.7 Economic effect

Assuming that annually produces four robots the total expenditure will be:

$$C_y = 4 * \text{Cost}_{\text{vat}} = 4 * 423372 = 1693488$$

Calculate the cost of your foreign analogue robot:

$$C_f = 4 * \text{Cost}_f = 4 * 1006000 = 4024000$$

Where C_f - price per 1 unit of foreign counterpart including postage, C_f = 1006000 tg

To determine the annual economic effect achieved through the introduction of the project calculate the cost savings of resources:

$$E = C_{fc} + C_y$$

Where, E - economic effect;

C_{fc} - the cost of foreign analogue (4024000 tenge);

C_y - the projected cost options (1693488 tenge).

E = 4024000 – 1693488 = 2330512

Payback period of the project calculated by the formula:

$$T = C_y / E,$$

where T - payback period, years

Thus, we can calculate the payback period developed an automated system using the formula:

$$T = 4024000 / 2330512 = 1,73$$

Based on the above calculations, we can say that the development of robots has decreased in cost. Because foreign robots cost several tens of times more expensive.

7 OCCUPATIONAL HEALTH AND SAFETY

Safety at work - the state of working conditions under which the possible impact on the working of harmful factors. Labor Protection - a system to preserve the life and health of employees in the course of employment, which includes the legal, socio-economic, and organizational - technical, sanitary, medical and preventive treatment, rehabilitation and other measures.

The problem of labor Protection is to minimize the risk of infestation or disease running at the same time providing comfort with maximum productivity.

The law of the Labor Protection reflects the following rules and regulations: rules for organizations from enterprises, regulations on accident prevention and health regulations, rules that ensure the protection of individual workers from occupational diseases, rules and regulations of the special protection of women, youth and persons with reduced working capacity; law, which provides for liability for violation of legislation on labor protection.

Providing healthy and safe working conditions imposed on the administration of the enterprise. Administration of the enterprise is obliged to introduce modern means of safety to ensure sanitary conditions and prevent occupational diseases of workers.

In aims of accident prevention AGB imposes on the administration of the company following functions: briefing on accident prevention, industrial hygiene and fire safety.

There are several types of instruction: introductory, primary in the workplace, secondary, unplanned, current. Induction training required to pass all newly arriving at the company, as well as seconded person. Primary workplace conducted with all who joined. Secondary - not less than six months. His goal - restoring the memory of the working rules for accident prevention, as well as parsing of specific violations.

Unscheduled accident prevention done when a change process, rules of labor protection or the introduction of new technology.

Current instruction is carried out by employees before work, which is made the admission in the outfit.

It is also necessary in offices in designated places to hang out stands with the rules of accident prevention instructions.

7.1 Safety requirements of workspace

Building (construction), which posted the jobs on the constitution must be consistent with their functional purpose and requirements of occupational health and safety. Working equipment must meet safety standards established for this type of equipment, have appropriate warning signs and provided with guards or protective devices to ensure safety of employees in the workplace. Emergency routes and exits of employees from the premises must be left open and display the open air or in the safe zone. Hazardous areas must be clearly marked. If the jobs are located in hazardous areas, which due to the nature of the risk to the employee or falling objects, such places should be equipped with devices as possible, barring access to these areas to outsiders. Pedestrians on the territory of the organization and technology, vehicles must travel in a safe environment. Workers should have the means of individual protection for work in hazardous industrial facilities (stations), including height, ground conditions, the open cells on the shelf seas and inland waters. During working hours, temperature, natural and artificial lighting and ventilation in the room where the jobs are located, must comply with safe working conditions. Employees are allowed to work in hazardous conditions (dust, fumes and other factors) after providing the employer safe working conditions.

Meteorological conditions (or climate) in the production are determined by the following parameters: air temperature, relative humidity, air velocity and pressure. However, on human health is strongly influenced by pressure differences. The quantity of heat Q by the human body depends on the degree of stress in certain circumstances and may range from 80 J / s (at rest) to 500 J / s (hard work).

With environmental impact of heat convection and radiation is terminated. With decreasing t Environment blood vessels narrow and blood flow to the surface of the body slows down and the heat transfer decreases. C. At high temperatures, room air blood vessels dilate□Humidity affects the body's thermoregulation: high humidity (over 85%) makes it difficult to regulate its temperature by reducing the evaporation of sweat, and too low (less than 20%) - causes drying of the mucous membrane solid resulting in increased blood flow to the body surface and heat transfer in the environment grows. The optimum value of moisture content of 40-60%. Air movement has a big impact on well-being. In winter, the air velocity should not exceed 0,2-0,5 m / s, and in summer - 0,2-1 m / sec. The velocity of the air may have an adverse impact on the spread of harmful substances.

The desired composition of the air can be achieved by performing the following activities:

1) The application of technological processes and equipment, excluding the formation of harmful substances. Of great importance is sealing equipment, which contains harmful substances;

2) Protection of sources of thermal radiation;

3) The device heating and ventilation;

4) The use of personal protective equipment.

7.2 Classification of ventilation systems

The objective of ventilation is to provide cleaner air in the given weather conditions. By way of moving air ventilation is natural and mechanical.

The ventilation system should not create noise in the workplace. It must be electric and non-hazardous.

Natural ventilation. Ventilation with natural ventilation is due to the temperature difference between air inside and outside, which causes flow of cold air into the room. With the dried up part of the building creates a low pressure, so that is extract warm air pollution from the premises. On the windward side of the building creates excessive pressure, resulting in the fresh air enters the room. Natural ventilation can be organized and disorganized. Unorganized ventilation is through leaks windows, air vents and special openings. Organized is maintained through natural ventilation and aeration vents.

Air conditioning. Air-conditioning - automatic maintenance of the premises, regardless of external conditions of temperature, humidity, cleanliness and air velocity. Air conditioning is used to create the necessary sanitation facilities. Air Conditioning - air handling unit, which with the help of instruments of auto regulation maintains indoor air quality parameters which are defined.

Electrostatic precipitators. Used for air purification from dust and mist. For medium and fine air filters are used, in which dust-laden air is passed through a porous filter material. Deposition of solid and fatty particles on filter elements occurs through contact with the surface of the particle pores. The mechanism of particle deposition, caused by the action of inertial forces, gravitational forces, the Brownian diffusion of gases and the effect of touch. As the filter material fabric, felt, paper, steel wool, porous ceramics and porous metals are used.

7.3 Protection against fire and explosion

The process of combustion is divided into several types: flash, fire, ignition, spontaneous combustion and explosion.

Activities for fire prevention are divided into organizational, technical and operational. Organizational measures include proper maintenance of machinery, proper maintenance of buildings and fire drill workers and employees. The technical activities include the observance of fire regulations, rules for the design of buildings, device wiring, heating, ventilation and lighting. Activities of the regime character - the prohibition of smoking in prohibited places, the production of welded and hot work in the fire areas. Operational measures - preventive check-ups, repair and testing of process equipment.

Apparatus for extinguishing fires. For fire fighting use fire extinguishers or portable installations. By hand extinguishers are foam, carbon dioxide, carbonic acid and powder.

Automatic fire alarms, depending on the influencing factors are smoke, heat and light. Smoke factor responds to the appearance of smoke. Heat to raise the room temperature. Light - light on an open flame. Heat detectors, automatic type used by the sensing element can be divided into bimetallic thermocouple and semiconductor.

7.4 Lighting

When light industrial buildings use natural light, artificial, carried out bulbs and combined. Natural light is divided into a lateral (through windows), upper (via aeration lights, ceiling openings), combined.

Artificial light can be of two kinds: general and combined. General lighting is even without taking into account the location of the object and the total localized taking into account the location of jobs. Application of a local lighting inside the building is not allowed. For administrative and storage facilities can be used for general lighting system. By their functional artificial lighting divided: business, emergency, evacuation, safety and rescue.

Requirements for industrial lighting. The main task lighting is to create the best conditions for the review of the object. This problem can be solved by the lighting system that meets the following requirements:

1) The illumination shall conform to the visual work, which is determined by the following parameters:

2) The need to ensure uniform brightness distribution of the working surface, as well as within the surrounding area;

3) In the field of view, there should be no direct or reflected on sparkle. Sparkle - increased brightness of the illuminated surface;

4) The value of illumination should be constant over time. This is achieved using stabilizing devices;

5) To select the optimal orientation of the light flux;

6) Is necessary to choose the spectral composition of light;

7) All the elements of lighting installations, step-down transformers, must be durable, and power-, explosion- and fireproof.

Rationing of artificial lighting. The existing industrial lighting standards are defined as quantitative (minimum light), and qualitative characteristics (glare, ripple depth). Value of the minimum illumination is set by the characteristics of visual work, which define the smallest object size differences, the contrast of the object with the background and characteristics of the background. Distinguish between 8 bit and 4 subclass of work depending on eyestrain.

Normalization of natural light. Natural light is characterized by the fact that the emerging light varies widely, depending on time of year, day, and weather conditions. Therefore, natural lighting can not be defined quantitatively. As a normalized value for natural lighting using daylight factor (RED), which represents the ratio of light at a given point inside the building to the value of outdoor lighting, light produced by a fully open the skies. Normalized values of this coefficient are determined by the table with regard to the nature of visual work, lighting systems, and the area ofthe object. In addition to quantitative indicators KEO use qualitative indicator - the unevenness of natural light.

7.5 Protection from electromagnetic microwave radiation

When you use the computer, around some of its parts creates an electromagnetic field that creates the effect of microwaves, negatively acting on any living organisms. Millimeter waves are absorbed by the surface layers of human skin, centimeter - the skin and the adjacent tissues, UHF - penetrate to a depth of 10 cm. That is, electromagnetic waves can pass right through a person. And it is not so harmless. The electromagnetic field affects human cells and tissues, leading to violations of conditioned reflex activity, reduce brain bioelectrical activity and changes in inter-neuronal connections. Usually this is manifested by headache, fatigue, deteriorating health, hypotension, changes in the conductivity of the heart muscle.

Influencing the person, the electromagnetic field increases the temperature of his body, which leads to the selective heating of tissues and organs. The most vulnerable in this case, the following organs of the human body: the liver, pancreas, bladder, stomach. Their heat can easily exacerbate chronic illnesses (ulcers, bleeding, and perforation).

To improve safety when working at a computer must first of all draw attention to the elaboration of an ergonomic workplace. And buying an ergonomic keyboard can not do. To improve the ergonomics of the workplace must take the following steps.

Position the monitor so that its top spot was right before your eyes, or above, which will keep your head straight, and eliminate the development of cervical degenerative disc disease. The distance from the monitor to eye must be at least 45 cm.

Prevention of respiratory diseases include wet cleaning and airing the room. Nice place near the computer's aquarium, it will increase the humidity, to the same fish calms the nerves.

To mitigate the effects of electromagnetic fields, to reduce air deionization, the violation of air supply balance in a room with a computer air ionizer. Ions beneficial effects on people: improving mental and physical condition, increased disease resistance, reduced the number of bacteria in indoor air is cleared of suspended particulates, attenuated the effect caused by static electricity.

7.6 Guidance of workers on occupational health and safety

Training, coaching, validation of knowledge workers on occupational health and safety held by the employer at their own expense. The order and timing of training, coaching and testing of knowledge on safety and health of workers shall be determined by state labour body in coordination with other competent government authorities of the respective spheres of activity. Persons recruited to work on a mandatory basis are organized by the employer prior training with the subsequent conduct of mandatory testing of knowledge on health and safety. Employees who have not passed the preliminary training, guidance and validation of knowledge on health and safety, the work is not allowed. Executives and individuals, industrial organizations, responsible for ensuring the health and safety, periodically, at least once every three years, must undergo training and examination for the occupational health and safety training courses in their respective educational institutions.

In conclusion, labor employment rights takes place in a particular work environment, which for non-compliance with hygiene requirements may adversely affect to performance and health.

CONCLUSION

In this thesis we have discussed obstacle detection and avoidance for an autonomous farm tractor. We initiated the discussion by examining the motivation for the work. We then presented the tractor project as a whole, detailing how each of the subsystems worked together and then describing each subsystem separately. We next turned our attention to obstacle detection technology, describing the different sensors used in literature as well as the sensor selected to be used for the project. We then discussed obstacle avoidance. We highlighted the differences between global and local avoidance and we specified five criteria for an avoidance algorithm that would merge into our design architecture. We then discussed different avoidance algorithm and considered if they could be merged into our design. The details of the algorithm that fit our design were then presented. We then discussed the obstacle detection and avoidance system that was used on the tractor. We highlighted the differences between our avoidance algorithm and Borenstein's algorithm. Finally we presented the simulation and testing procedures that helped us refine our avoidance algorithm. Now that all is said and done, what have we learned and where can this work lead us? In retrospect the avoidance algorithm we ended up using was very simple, yet the path to simplicity is often steep and treacherous. Many times as engineers we tend to complicate matters when the simplest answer is often the best. We experienced this more than once on the project. Another lesson learned was that man in an incredible machine. We take for granted the powerful sensors our eyes truly are and we hope that one day82 some genius will be able to construct a sensor as effective as the eye. The detectionsensor is truly the limiting factor in this whole process. No matter how clever the whole system is, without the knowledge of the surrounding environment we cannot operate safely in any farming environment. Fortunately there are clever people who are trying to overcome this obstacle with the technology of today. By using a combination of different detection sensors and fusing the data a solution may be possible. Where will this work lead in the future? As was demonstrated in this thesis, the author examined many different obstacle avoidance techniques and then designed a unique approach based loosely on other people's work. The author's avoidance algorithm, while different than all other avoidance algorithms, is specific to the farming environment, specifically slow moving vehicles that do deviate much from the desired path of travel because of possible crop damage. In order to improve on this algorithm more testing in the field needs to be done. More testing would allow us to try different exit angles from the path and different return angles to the path, finding the optimal solution. More testing would also demonstrate the robustness of the algorithm. The obstacle filter, which was a novel approach to combining global and local obstacle avoidance, can be improved to try and take out spurious data. Some improvements could be pattern recognition; examining the filter map and determining if a cluster of cells is dust (spurious data) or an actual obstacle. This avoidance algorithm could also be used in other autonomous vehicle research, such as the omni-directional vehicles being developed by Kazakh British Technical University. The algorithm would have to be adapted to the specific robot function, but the basis for the algorithm stays the same.

References

1)Engelberger, J., Transitions Research Corporation, private communication, 1986.

2) Borenstein, J., "The Nursing Robot System." Ph. D. Thesis, Technion, Haifa, Israel, 1987.

3)Borenstein, J. and Koren, Y., "Obstacle Avoidance With Ultrasonic Sensors." IEEE Journal of Robotics and Automation. Vol. RA-4, No. 2, 1988, pp. 213-218.

4) Borenstein, J. and Koren, Y., "Real-time Obstacle Avoidance for Fast Mobile Robots." IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept./Oct. 1989, pp. 1179-1187.

5) Paul Kinsky and Quan Zhou, “Obstacle Avoidance Robot.” Major Qualifying Project Report, WORCESTER POLYTECHNIC INSTITUTE, Project Number: YR-11E1.

6) Ibrahim KAMAL, “Infra-Red Proximity Sensor Part 1.” Article, Sensor & Measurement, January 22, 2008. Available: http://www.ikalogic.com

7) Harivardhagini S., Pranavanand S and Ghali Bharadwaja Sharma, “Voice Guided Robot Using LabVIEW.” CVR Journal of Science and Technology, Volume 3, December 2012.

8) Jnaneshwar Das, Mrinal Kalakrishnan and Sharad Nagappa, “Obstacle Avoidance for a Mobile Exploration Robot with Onboard Embedded Ultrasonic Range Sensor”, PES Institute of Technology, Bangalore.

9) Frank Rubin.The Lee path connection algorithm IEEE Transactionson Computers. 1974. Vol. C-23. № 9. PP. 907–914.

10. Kochetkov S.A., Utkin V.A. A trajectory stabilization algorithm for mobile robot // The Proceedings of 11-th International Workshop on Variable Structure systems (VSS2010).

Mexico, 2010. PP. 121–127