Bachelor's thesis

Degree programme: Information Technology

Internet Technology

2014

Lok Prasad Khanal

OBSTACLE-AVOIDING ROBOT

A possible introduction project for engineering students

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR´S THESIS | ABSTRACT
TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme | Information Technology

December, 2013 | 50

Instructor: Patric Granholm

Lok Prasad Khanal

OBSTACLE-AVOIDING ROBOT

BOE-BOT, where BOE stands for Board-Of-Education, is a popular robot made by Parallax Inc. Board-Of-Education (BOE) is used in projects where it acts as a programmable intelligent board. The goal of this project is to get students interested in and excited about the fields of engineering, mechatronics, and software development as they design, construct, and program an autonomous robot. The thesis not only gives detailed information about Arduino and the use of App Inventor for android application design but also about the IR, IR sensors, PBASIC programming platform to the new students or beginners. The guidelines provided are very simple to use and understand thus, making it very easy for the new students to build a foundation in their Robotics learning.

There are two steps of the thesis; first, to assemble a robot whose main task is to detect the obstacles on its way and move away avoiding them if there are any, and thus, making its own obstacle free path, and second, introduce new students with the topics of robotics, Infrared Sensors, Arduino, Embedded Systems along with the use of robotics in the real life. The main aim of the project is to provide simple guidelines to the new students and beginners who are interested in this type of project. This will make students gain basic knowledge and skills regarding servo, program and mathematics to calculate program value.

Arduino can simply be defined as single-board microcontroller, intended to make the application of interactive objects or environments more accessible. The thesis will make new students familiar about the robotics world and their use in the real life. This project will help new students get familiar with Infrared Sensors, Microcontroller, Arduino, and thus providing a guideline to those students to make a mobile robot that might have a real significant use in the world.

KEYWORDS:
Arduino, Embedded Computing, Robotics

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots. A machine capable of carrying out a complex series of actions automatically, esp. one programmable by a computers is defined as a robot. And, Obstacle avoidance refers to the ability of a robot to detect obstacles in its way if there are any and thus make its own obstacle free path.

The thesis deals with two steps; first making an obstacle avoiding robot and second, introductory guideline to the first year engineering students. The thesis will help them to learn about physics when dealing with terms like Infrared (IR), IR sensors, electromagnetic spectrum, and also with embedded computing while making the robot. The Board of Education (BOE-Bot) is our working basement of the project. BOE-Bot is relatively simple programmable robot series which does not require any deep knowledge of robotics, programming, or electronics. The project is to develop a robot that will move according to the code assigned but find a free space, navigating from any obstacle on its way. This kind of obstacle is very useful in industries where automated supervision is needed, for example, in places where it might be risky for humans to be. This robot can also be made by putting other sensors like light sensors or line sensors depending on the need. However, putting camera in the robot will make it a smart robot that this may help humans if needed. For example, it might not always be possible to go to every places but we can send this robot which will be there making its own path and send different information.

The project provides a guideline to the students who are new in the world of Arduino and help them to understand about embedded system, IR sensors, microcontroller and how to make a robot using Arduino. The thesis will make students learn more about basic knowledge and skills regarding servo, program and mathematics to calculate program values. New students will learn how to program the BOE-Bot to perform basic maneuvers and gradual acceleration and deceleration of the robot to get robot out of maneuvers and also students will learn to write subroutines to perform basic maneuvers.

The aim of the thesis is to evaluate what students can learn about the fields of engineering, mechatronics, and software development as they design, construct, and program an autonomous robot. The thesis not only gives detailed information about Arduino and the use of App Inventor for android application design but also about the IR, IR sensors, PBASIC programming platform to the new students or beginners. The guidelines provided are very simple to use and understand thus, making it very easy for the new students to build a foundation in their Robotics learning. This project is very much helpful to the first year students who have a keen desire and interest in the robotics and specially Arduino robotics. From this students will learn about servo motors and how to program them.

2. 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

2.1. 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).

2.2. 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 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)

2.3. 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 (Wheat, 2001).

2.4. 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 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.

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.

3. EMBEDDED SYSTEM (ES)

The microprocessor-based system is built for controlling a function or range of functions and is not designed to be programmed by the end user in the same way a PC is defined as an embed system. An embedded system is designed to perform one particular task albeit with different choices and options.

Embedded systems contain processing cores that are either microcontrollers or digital signal processors. Microcontrollers generally known as "chip", which may itself be packaged with other microcontrollers in a hybrid system of Application- Specific Integrated Circuit (ASIC). In general, input always comes from a detector or sensors in more specific word and meanwhile the output goes to the activator which may start or stop the operation of the machine or the operating system.

An embedded system is a combination of both hardware and software. Each embedded system is unique and the hardware is highly specialized in the application domain. Hardware consists of processors, microcontroller, IR sensors etc. On the other hand, Software are just like a brain of the whole embedded system as this consists of the programming languages used which make hardware work. As a result, embedded systems programming can be a widely varying experience.

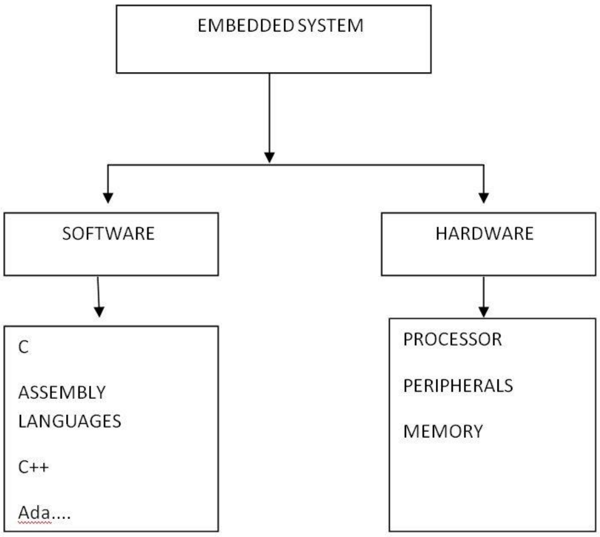


Figure 1. Block diagram for Embedded System (ES consists of hardware and software part which again consists of programming language and physical peripherals respectively).

If the history of ES is observed, it can be figured out that the grace of ES have been enjoyed for quite a long time since they were designed around the microprocessors than microcontrollers, which are used mostly today. There has been a huge shift in ES from microprocessors to microcontrollers because microprocessors do not possess any memory, ports etc., as a result, everything must be connected externally by using peripherals like 8255, 8257, 8259 etc. (Raviraj, 2009)

On the other hand, the microcontroller is a single silicon chip consisting of all input, output and peripherals on it. A single microcontroller has the following features:

1. Arithmetic and logic unit

2. Memory for storing program

3. EEPROM for nonvolatile and special function registers

4. Onput/output ports

6. Analog to digital converter

7. Circuits

8. Serial communication ports and many other

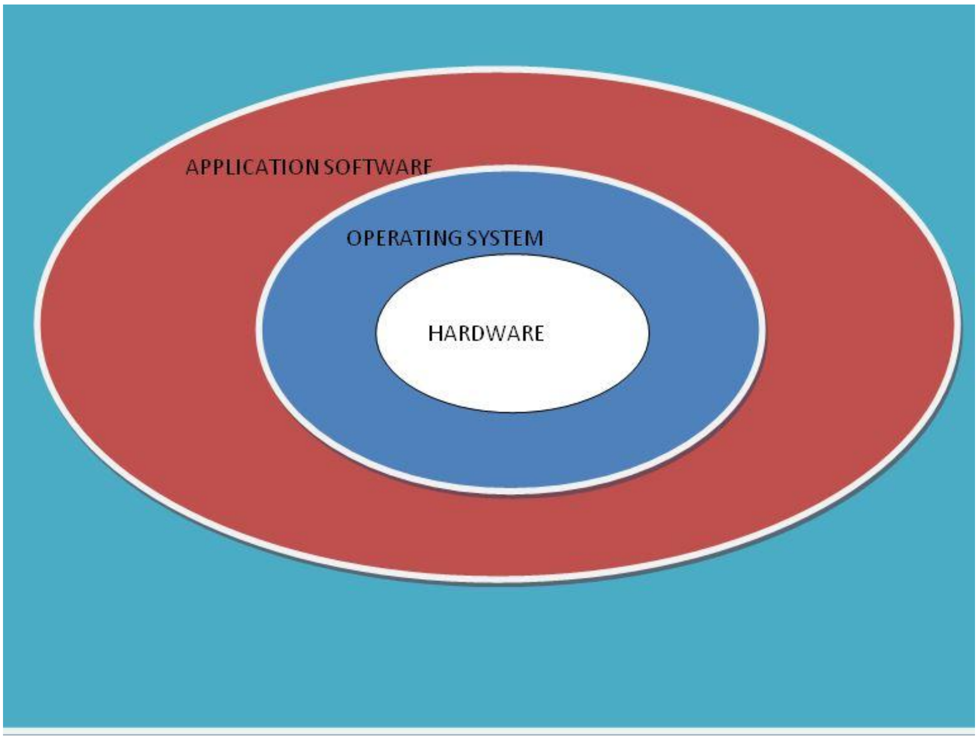


Figure 2. Layered architecture of an Embed System.

A microprocessor-based system which is built for controlling a function or range of functions and is not designed to be programmed by the end user in the same way a PC is defined as an embedded system. An embedded system is designed to perform one particular task albeit with different choices and options.

4. MICRO CONTROLLER

A highly integrated chip that contains all the components comprises a controller. Typically this includes a CPU, RAM, some form of ROM, I/O ports, and timers. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task; to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production costs.

Microcontrollers are sometimes called embedded microcontrollers. This just means that they are part of an embedded system; that is, one part of a larger device or system. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. The first integrated circuit was developed by Jack Kilby of Texas Instruments and Robert Noyce of Fairchild Semiconductor in 1950.

5. ROBOTICS

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots. A machine capable of carrying out a complex series of actions automatically, esp. one programmable by a computers is defined as a robot. Robotics must be able to perform certain tasks assigned, within given limitations, regardless of these limitations being human-controlled or automatic.

5.1 Robot

An electro-mechanical machine that can do the work of a person and that works automatically or is controlled by a computer is defined as a Robot. It is a device that can perform automatically or through some controlling devices. A robot is defined as " a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer.” (Oxford Dictionary, 2013)

5.1.1 What should a Robot look like?

According to the survey conducted by the Swiss Institute of Technology in Lausanne in 2008 among 240 people (Swiss Federal Institute of Technology, 2008), this is the general concept of what a robot should look like (Fig. 3):

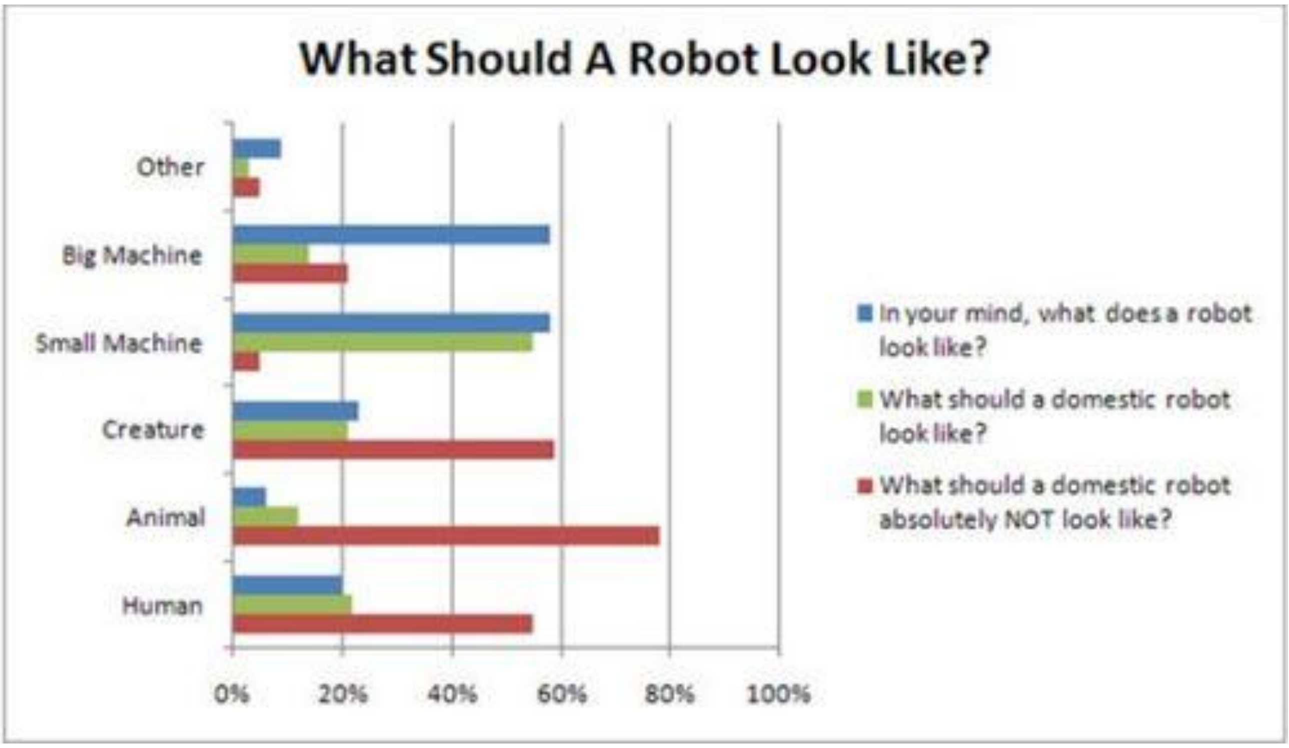


*Figure 3. What should a Robot look like? (*Swiss Federal Institute of Technology, 2008)*.*

Results show that for the participants in the survey a robot looks like a machine, be it big or small. In spite of the apparent popularity of Japanese robots such as the Sony Aibo, the Furby and Asimo, other categories (creature, human, and animal) gathered only a small percentage.

5.2. The robot task

The task of our robot is pretty simple. The robot has to be made in such a way that it will move forward until it detects an obstacle. On detecting an obstacle, it will turn left or right depending on the free space available in front of it. On detecting any obstacles through the IR sensors, it moves towards left or right depending on the amount of IR rays detected by IR sensors (left and right) respectively. Although the task is simple, this robot is completely autonomous. At the end of the project, the final robot looked like this (Fig. 4):

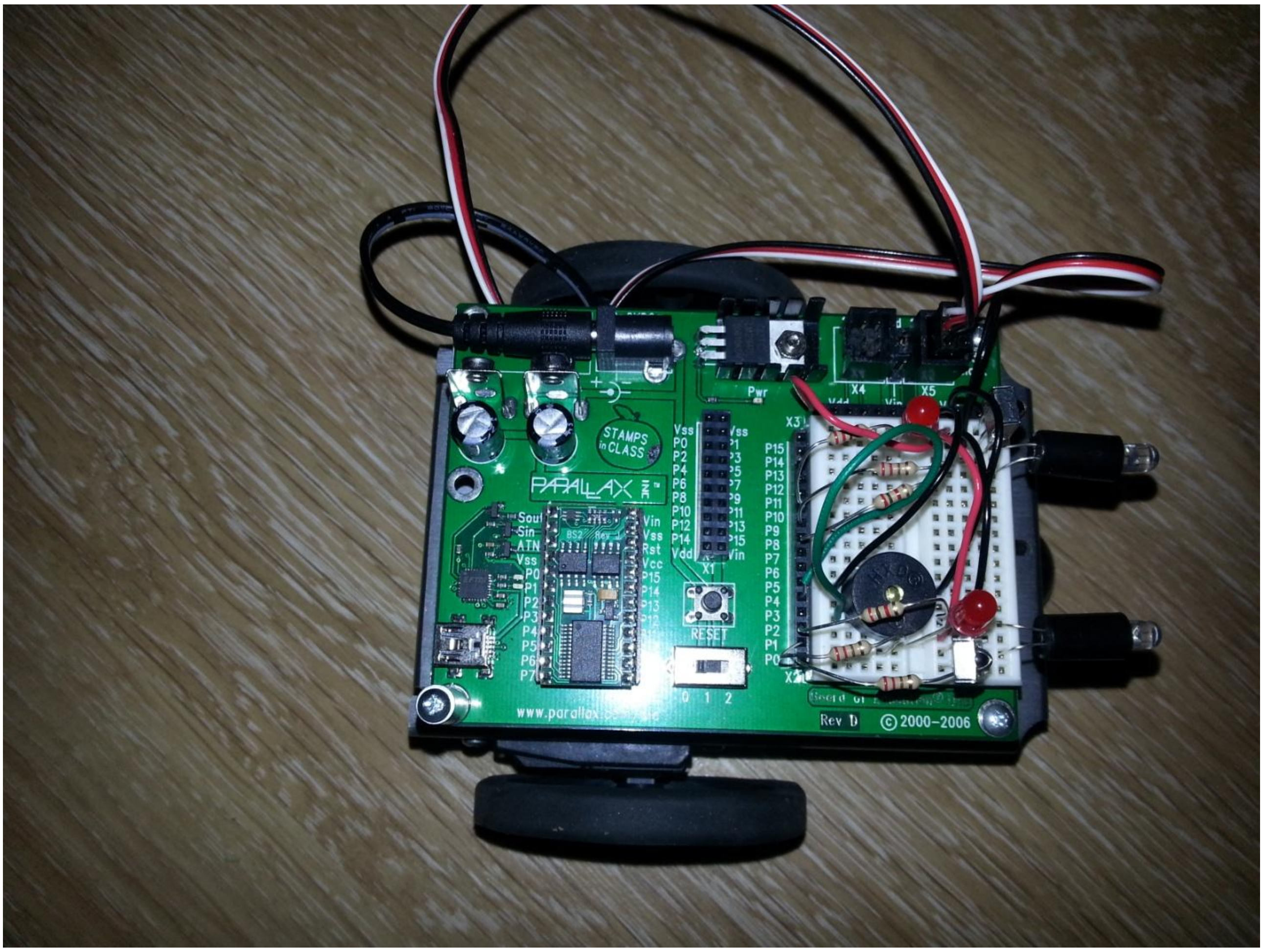


Figure 4. Final robot concept.

6. IR SENSORS

An infrared sensor is an electronic device that detects infrared radiation in order to sense some aspect of its surroundings. Infrared sensors can measure the heat of an object.

Infrared light is electromagnetic radiation with longer wavelengths than those of visible light, extending from the nominal red edge of the visible spectrum at 700nm to 0.1mm. This range of wavelengths corresponds to a frequency range of approximately 430 THz down to 300 GHz. In a motion detector, a emitter emits IR rays which gets reflected from obstacles on its way and the IR rays reaches the sensor itself at the center of the device. This part may be composed of more than one individual sensor, each of them being made from pyroelectric materials, whether natural or artificial. These are materials that generate an electrical voltage when heated or cooled.

These pyroelectric materials are integrated into a small circuit board. They are wired in such a way so that when the sensor detects an increase in the heat of a small part of its field of view, it will trigger the motion detector's alarm. It is very common for an infrared sensor to be integrated into motion detectors like those used as part of a residential or commercial security system.

Most motion detectors are fitted with a special type of lens, called a Fresnel lens, on the sensor face. A set of these lenses on a motion detector can focus light from many directions, giving the sensor a view of the whole area. Instead of Fresnel lenses, some motion detectors are fitted with small parabolic mirrors which serve the same purpose.

An infrared sensor can be thought of as a camera that briefly remembers how an area's infrared radiation appears. A sudden change in one area of the field of view, especially one that moves, will change the way electricity goes from the pyroelectric materials through the rest of the circuit. This will trigger the motion detector to activate an alarm. If the whole field of view changes temperature, this will not trigger the device. This makes it so that sudden flashes of light and natural changes in temperature do not activate the sensor and cause false alarms.

Infrared motion detectors used in residential security systems are also desensitized somewhat, with the goal of preventing false alarms. Typically, a motion detector like these will not register movement by any object weighing less than 40 pounds (18 kg). With this modification, household pets will be able to move freely around the house without their owners needing to worry about a false alarm. For households with large pets, sensors with an 80-pound (36 kg) allowance are also made.

The robot used in this project will have two IR sensor elements; each will be mounted on left and right of the robot. Figure 5 below shows the positions of the two sensors.

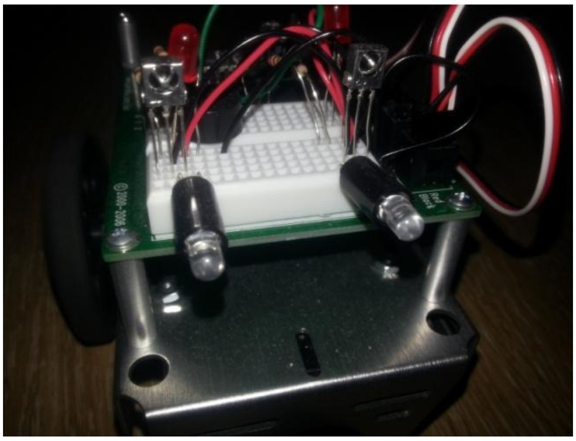


Figure 5. Position of sensors.

The whole sensor module shown above is made on a breadboard which can be bought from any electrical shop.

Infrared Light

The band of light which is a thousand times wider than that of a visible light is defined as Infrared light. Physically, light with wavelength from 0.7 µm to 0.1 mm is called Infrared Light. As our eyes cannot see the light with wavelength longer than 7×10^{-5} µm, thus IR light is invisible to the human eye. Beyond everything, infrared radiation is just like visible light as it possesses the same properties as visible light which can be focused and reflected as a normal visible light thus helping to detect it by the IR receiver in the project.

Because the Infrared light is not detectable for the human eyes, it is chosen for this purpose. The relation between frequency and wavelength is given by the equation:

$$f = v/\lambda$$

here,

f = Frequency

v = Speed of light c = 3×10^8 m/s

λ = Wavelength

Frequency is measured in Hertz = Hz = 1/seconds.

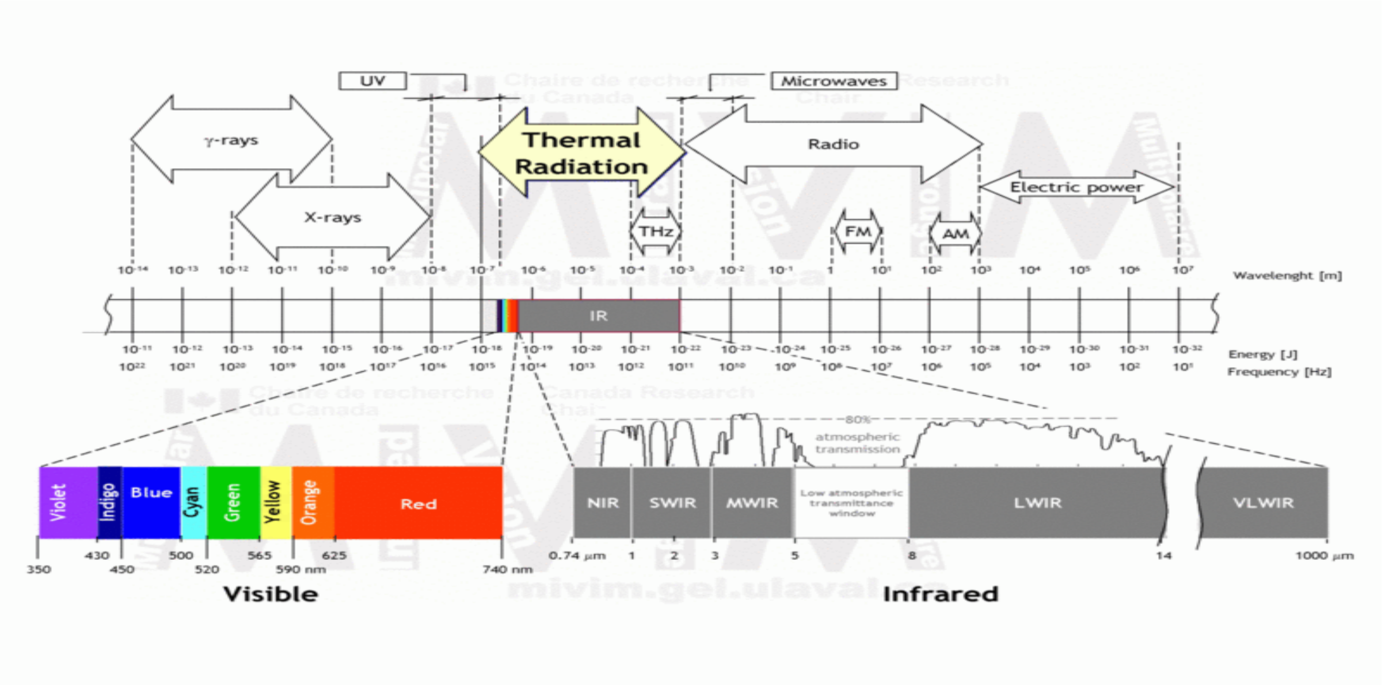


Figure 6. The infrared bands in the electromagnetic spectrum. (Griffiths,1999).

As infrared is precious in the near range distance which ranges up to 20cm, it will make the robot's movement more accurate.

7. Requirements for the BOE-Bot

Out of different microcontroller available, BASCI Stamp II (BS2), which is commonly known as BS2-IC, will be used in the project. Some of features of BS2-IC are listed below:

a. It can operate in high temperature ranging 40°C to 85°C.

b. It contains high number of instructions (500).

c. It operates at 20MHz executing approximately 4000 instructions per second.

d. It can recognize a total of 42 PBASIC commands.

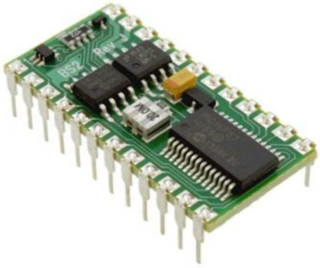


Figure 7. BS2-IC.

Board-Of-Education

The green detachable board mounted on the top of the chassis is called the board of education (BOE). BOE is an intelligent, smart programmable board developed by Parallax Inc. which accepts the BASIC Stamp controller chip. Because of its simplicity, affordability, large user base and easy for documentation, it is one of very popular boards. Figure 8 shows the board of education.

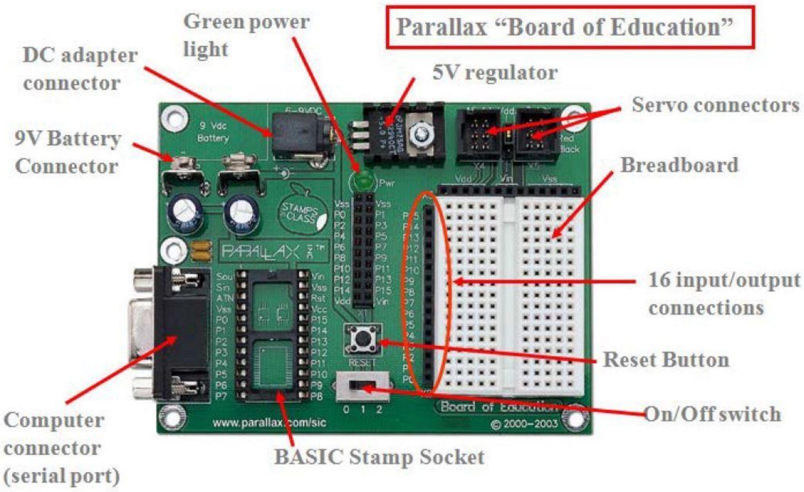


Figure 8. Board of Education.(Parallax, 2012)

BOE consists of many parts whereas the green base has paths for communication and in case of reset and on/off switch, paths can actually be seen because of its thickness which is meant for passing more current through. Different parts of the activity board are discussed below.

On/Off Switch

This switch consists of 3 settings:

- 1 --> Powers BOE excluding servos
- 2. --> Powers all components
- 3 --> Power down

Reset Bottom

This is made to restart the program. This basically disconnects power for a little while and reconnects again.

Green Power Light

This indicates the board is on if it is blinking and if not, it indicates there is no power on the board.

BASIC Stamp socket

It can also be defined as a meeting point of body and brain.

Computer connector

The USB port on the end of the activity board connects the BOE-Bot robot to the computer to execute program.

Input/output Connections

These are used to connect anything to the BOE-Bot just like power. You have to connect resistors to these pins and then to the breadboard.

Breadboard

A breadboard is a construction base that is meant for a one of a kind electronic circuit known as a prototype. It does not require soldering hence is reusable. This makes it easy to create temporary ones especially when carrying out experiments.

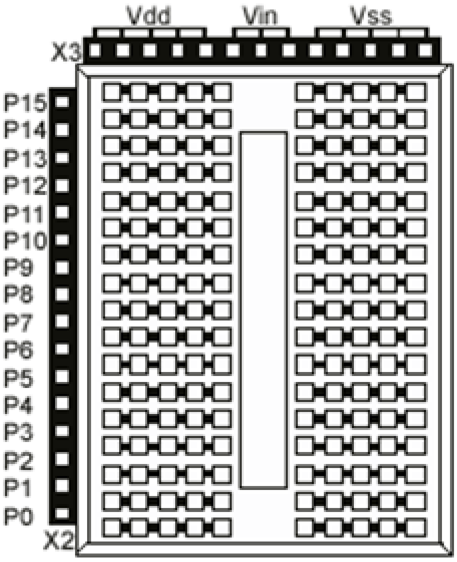


Figure 9. Breadboard.

Vdd, Vin, Vss

- **Vdd**: Voltage Input. Vdd is Drain-Drain voltage
- **Vin** : the voltage input from power supply to the voltage regulator on the board that is used to regulate Vin down to VDD for the chip which can be a range of voltages that regulator can tolerate.
- **Vss** : Vss refers to source-substrate voltage. It is a common ground (0 volts),

Chassis and other required parts

This section discusses the chassis and other parts required for the project.. Having discussed the very important required components of for the project, here are listed a few more items which are used:

- 2 Plastic wheels and band tires
- Tail wheel
- Cotter pin (1/16*)
- Red LEDs
- 2 Infrared detectors
- 2 Infrared LEDs
- Piezospeaker
- Jumper wires
- Resistors Parallax continuous rotation servos

- Parallax screwdriver
- pan head screws
- zinc-plated nuts
- rubber grommet
- standoffs and spaces

8. ASSEMBLING

8.1 Mechanical Assembly

The first task was to gather the required kits for the development of our proposed obstacle-avoiding robot. There are many online stores from where we can purchase all the kits, separate or as a whole pack at once. The Parallax website (http://parallax.com/) is one of the most renowned websites selling all types of products and services for the robotics.

The first task needed to be done was to assemble the parts starting from mechanical and the electrical components and then do the programming.

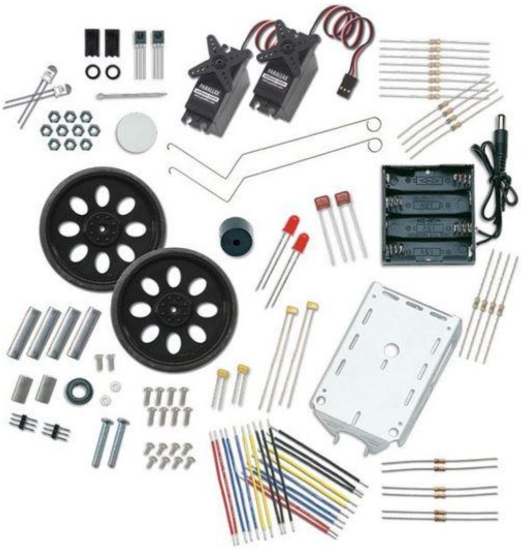


Figure 10. Parts included in BOE-Bot Robot kit.

8.1.2 Process

The process includes of mechanical construction of the robot followed by circuit connection. Mechanical construction represents the process of assembling the chassis whereas circuit connection includes the circuit design of the robot. Figure 11-16 below show the process of assembling the chassis:

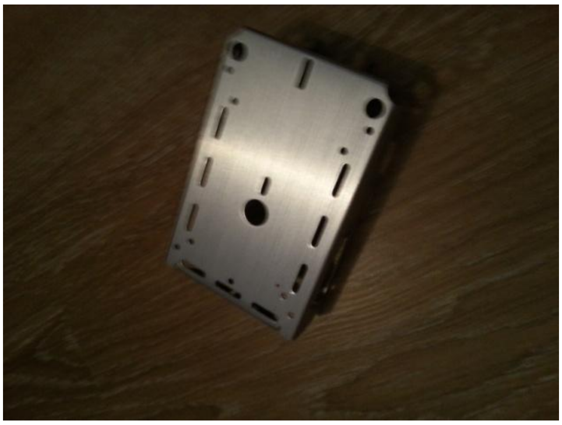


Figure 11. Chassis.

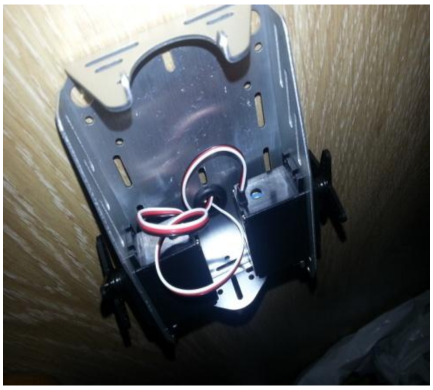


Figure 12. After mounting left and right servos on the chassis.

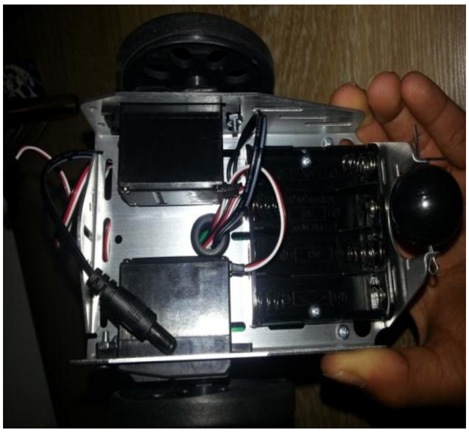


Figure 13. Battery holder.

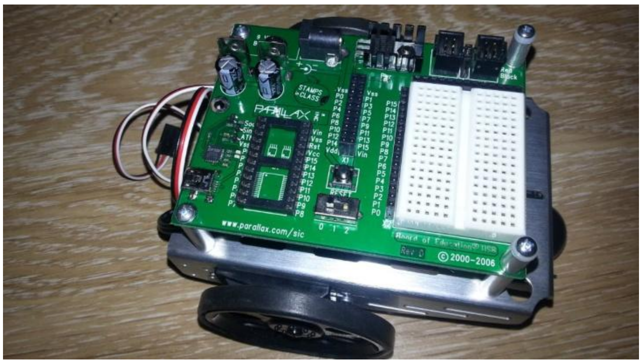


Figure 14. Activity board attached.

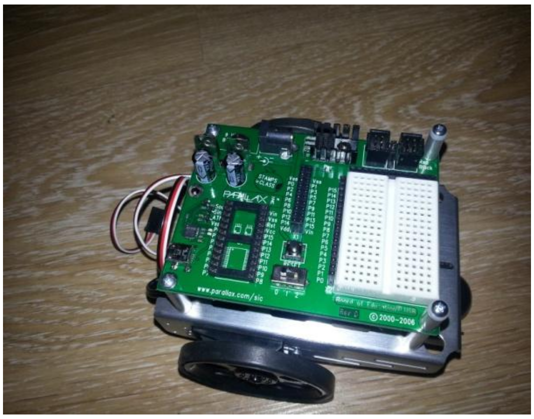


Figure 15.Aaerial view of the chassis.

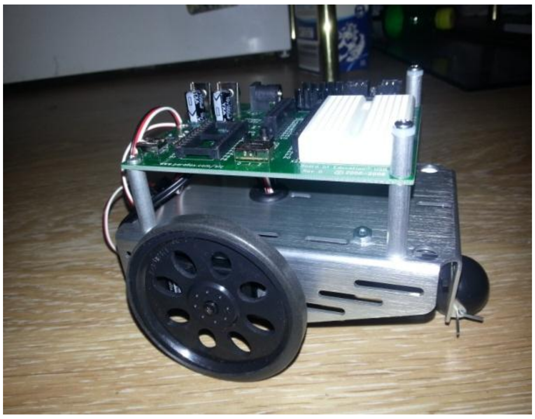


Figure 16. Side view of the chassis.

8.2 Electrical Connections

Before steeping into the world of electrical connections, programming and the real challenging practical part of the project, we needto know the logic of the brain of the Parallax, Inc's Boe-Bot® robot. It is a BASIC Stamp® 2 programmable microcontroller (Fig.17) which is not only powerful but also easy to use.

Figure 17. BASIC Stamp II.

The main tasks of the BASIC Stamp are to:

- monitor sensors to detect the world around it
- make decisions based on what it senses
- exchange information between Robot and Roboticist acting like an information exchange center (User)
- control the motion of the robot by operating motors which makes the wheel of the robot turn and make a move

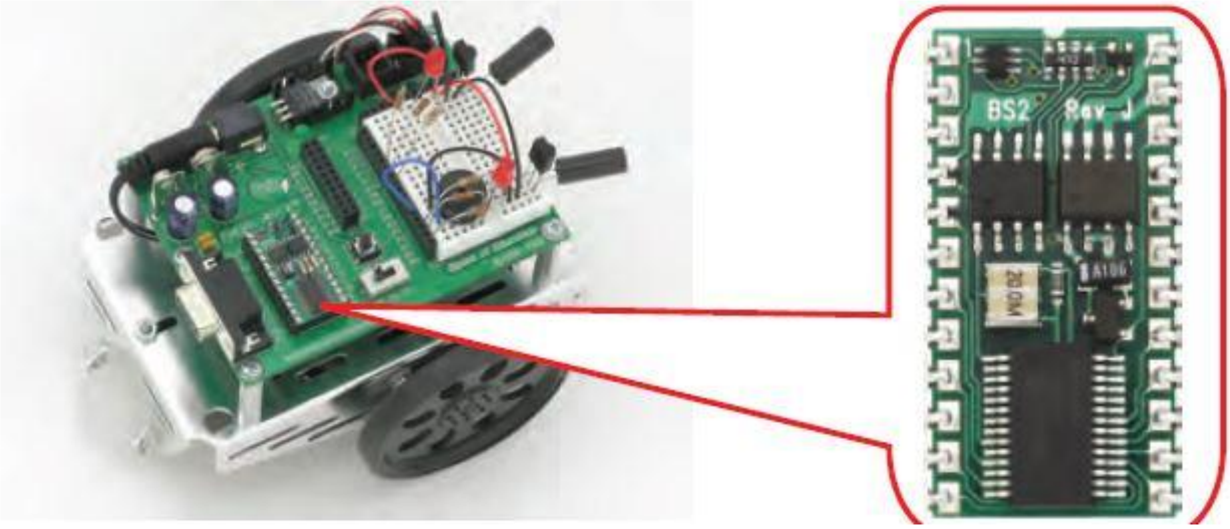


Figure 18. BASIC Stamp Module on a Boe-Bot Robot.

The electrical connection was started with attachment of the BASIC Stamp module in BASIC Stamp socket in Board of education. After the chassis was ready for the electrical connection, it was necessary check was if the red led was working well. First, the led was connected to the breadboard as shown in Fig. 27. After that, it was tested in the robot. Tthe HIGH and LOW commands can be used to make the BASIC Stamp connect an LED alternately to VDD and Vss.

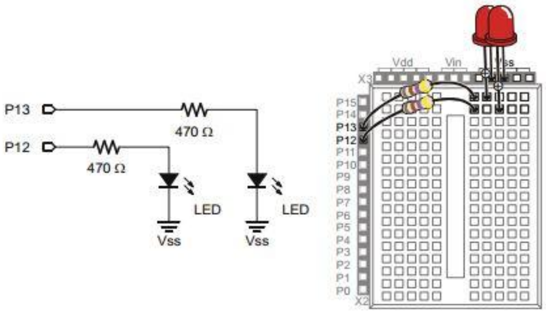


Figure 19. Schematic (left) and wiring diagram (right).

Here, in the coding above tells HIGH 15 which means that the BASIC Stamp to connect I/O pin P15 to Vdd, which turns the LED on.

Turn the LED connected to P4 on/off once every second.

```
{$STAMP BS2}

{$PBASIC 2.5}

DEBUG "The LED connected to Pin 13 is blinking!"

DO

            HIGH 15

           PAUSE 500

            LOW 13

           PAUSE 500

LOOP
```

Likewise, if we use the command LOW 13, then it means that the BASIC Stamp connects I/O pin P13 to Vss, which turns the LED off.

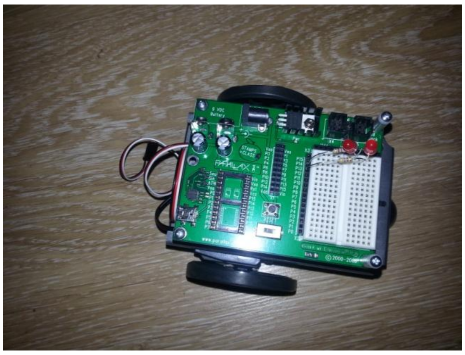


Figure 20. Testing LEDs.

After coding and compiling the code, if the red led start flashing, it indicates the fine connection of the circuit and BASIC Stamp editor (v2.5.3)

Then a device called piezoelectric speaker (piezospeaker) (Figure.21) was added. Its main task is to generating tones depending on the frequency of high/low signals it receives from the BASIC Stamp.

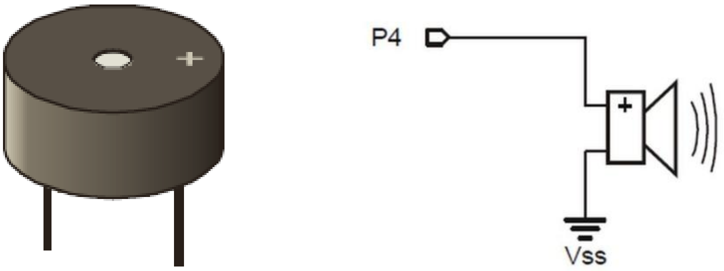


Figure 21. Piezospeaker.

The piezoelectric element is a crystal that changes its shape slightly when voltage is applied and upon the high and low voltage applied at a rapid rate, piezoelectric crystal changes rapidly in shape with vibration resulting in the production of the sound due to the vibration of the air caused by piezospeaker. The connection of piezospeaker on the robot is shown in Figure.22.

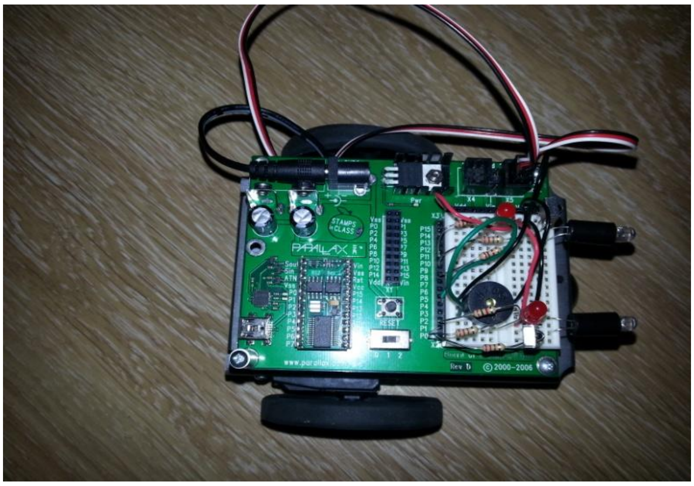
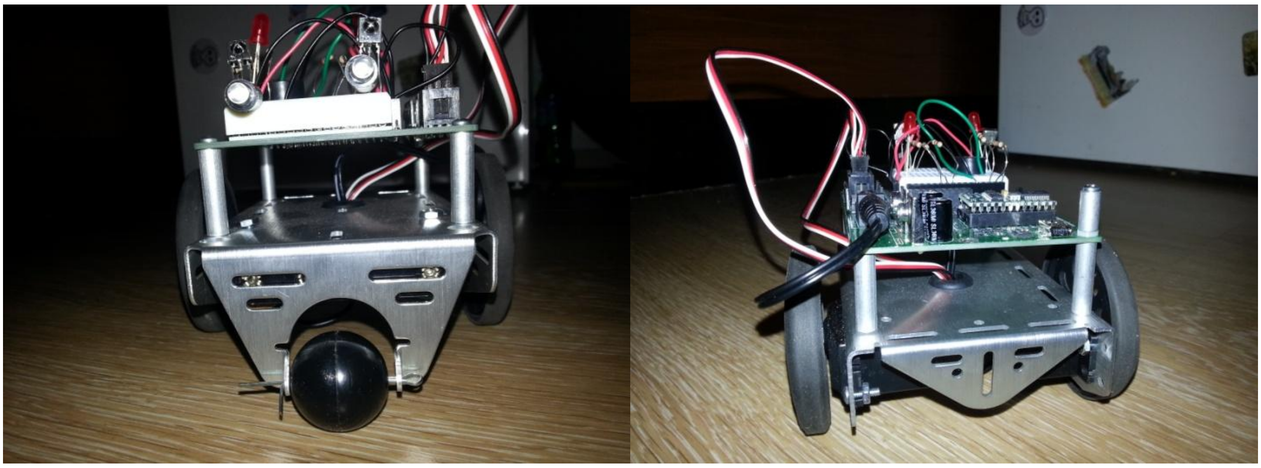


Figure 22. Position of the Piezospeaker.

After that the resistors, the jumpers and IR led were connected followed by the power connection. The final connection of the Robot are shown in Figures 23(a) and (b) below:



(a) (b)

Figure 23. Front view (a) and rare view (b) of the Robot.

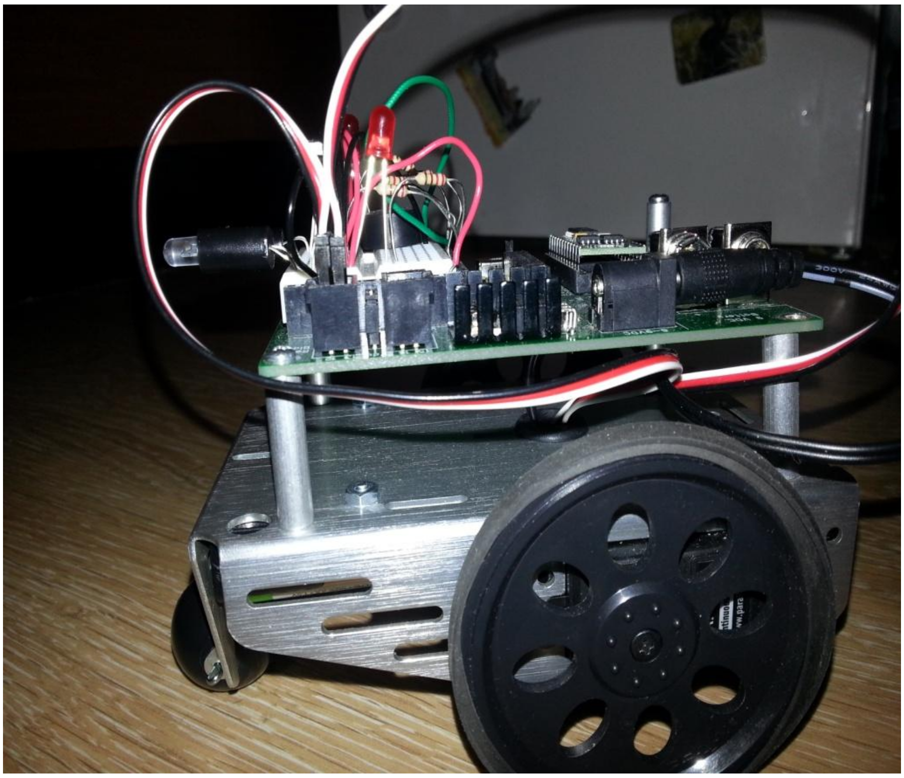


Figure 24. side view of the Robot.

After the mechanical and electrical connection was followed by the installing of the required software (BASIC Stamp Editor v2.5.3, the final product was ready to be coded and tested for programming. The final product of the project is shown in Figure 25 below.

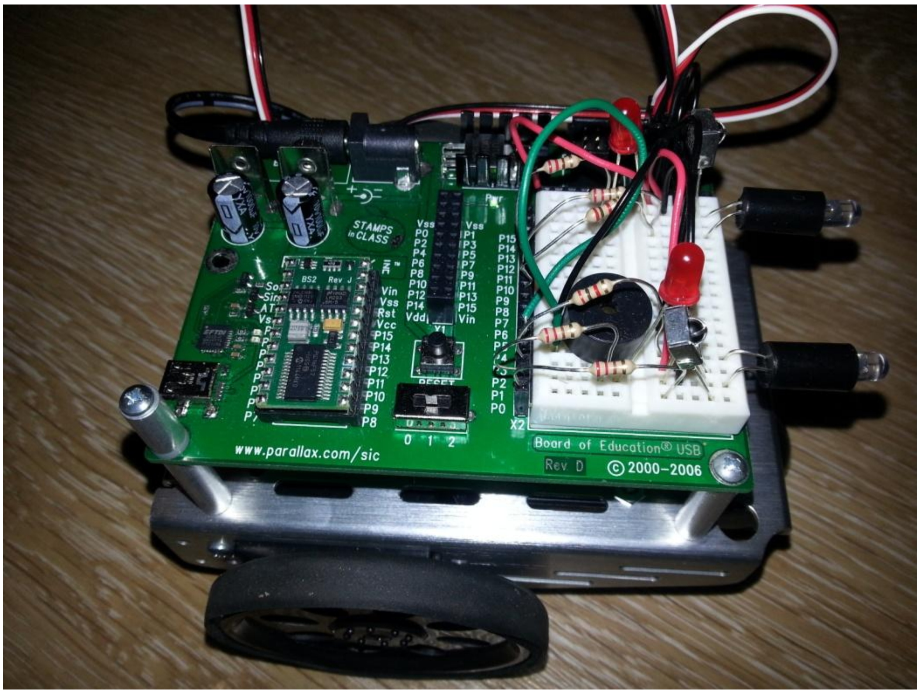


Figure 25. Boe-Bot Robot (final).

9. CONTROL

The servos rotate in one direction when the power is supplied to its terminal. However, if the power is supplied into the reverse polarity, the robot will move to the opposite direction.

An example of driving motors by using L298 is given below:

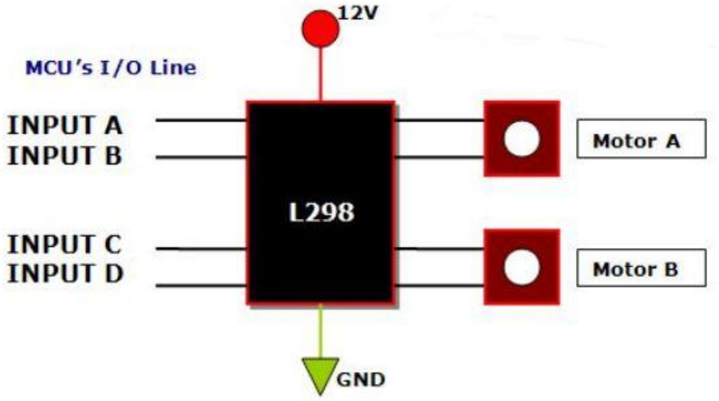


Figure 26. Basic L298 connection.

The driving system in the project is differential driving system. In this system, the wheels are rotated at different speeds resulting in the robot rotating right or left. That is why it is called Differential Drive. For example, the robot will turn right if the left wheel rotates faster than right wheel.

For this robot, we will be using the following rotation of wheel for steering and straight motion:

Table 1. Differential Drive System.

MOTION	RIGHT Wheel	Left Wheel
FORWARD	Clockwise	Counter Clockwise
BACKWARD	Counter Clockwise	Clockwise
Rotate RIGHT	Counter Clockwise	Counter Clockwise
Rotate LEFT	Clockwise	Clockwise

So moving and steering the Robot is just a matter of controlling the two direct current (DC) motors of the robot.

Motor A

- **Clock Wise(CW) :** Input A = **High** and Input B is **Low**
- **Counter Clock Wise(CCW) :** Input A = **Low** and Input B is **High**
- **Stop:** Input A= **High** and Input B is **High**
- **Stop:** Input B=**Low** and Input B is **Low**

Motor B

- **Clock Wise(CW) :** Input C= **High** and Input D is **Low**
- **Counter Clock Wise(CCW) :** Input C = **Low** and Input D is **High**
- **Stop:** Input C= **High** and Input D is **High**
- **Stop:** Input C=**Low** and Input D is **Low**

10. SCHEMATIC DIAGRAM

10.1 Board Of Education (BOE)

BOE is an intelligent, smart programmable board developed by Parallax Inc. which accepts the BASIC Stamp controller chip. Because of its simplicity, affordability, large user base and easy for documentation, it is one of very popular boards.

In Figure 27, 100nF capacitors is adjusted as close to the ICs power supply pins as possible which also applies to piezoelectric crystals, which are to put as close as possible to the XTAL pins of the ICs. The LM317 regulator is used to generate the necessary 3.3Volts for the Bluetooth module. If low-dropout voltage regulator LM2940 is not available, it can also be replaced with standard 7805 regulator.

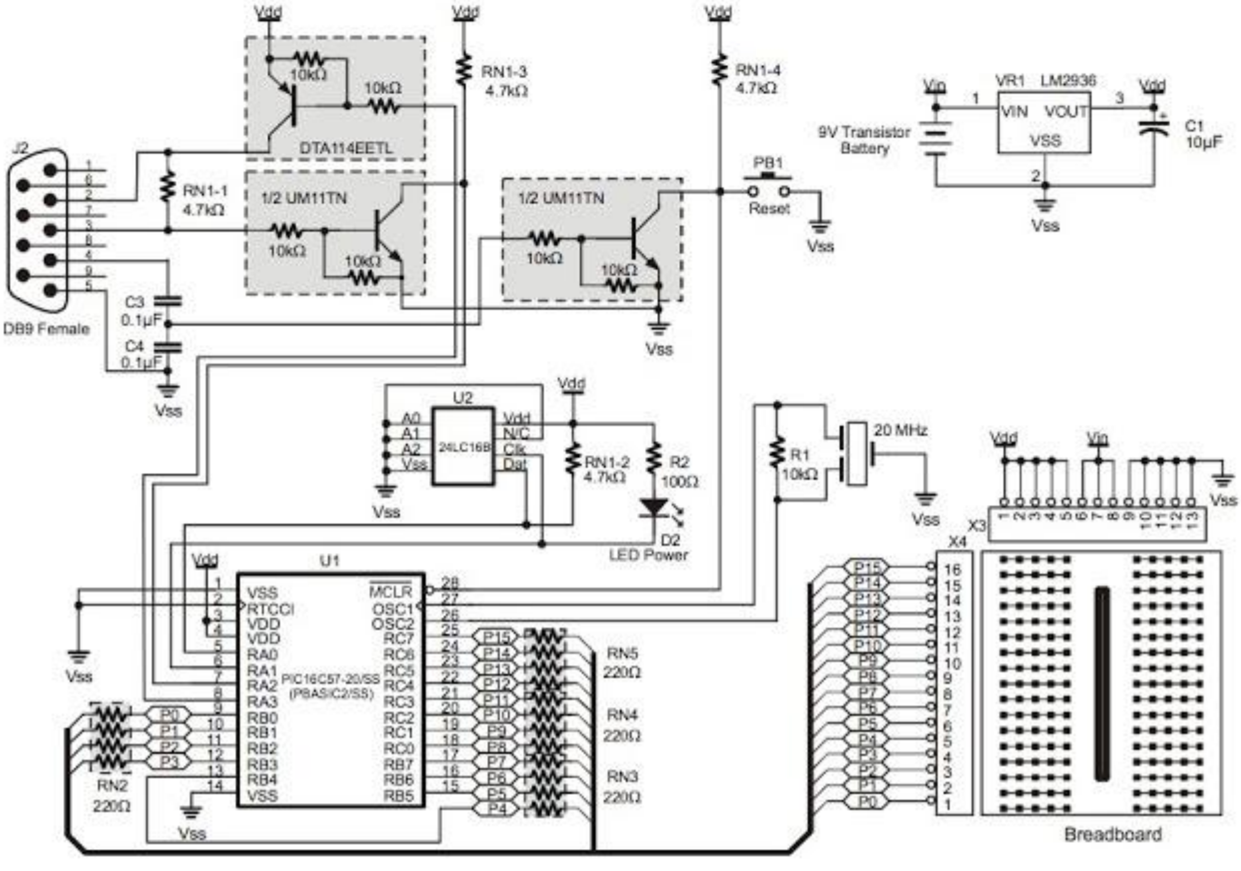


Figure 27. Schematic diagram of Board of Education.

10.2 BASIC Stamp II

The BASIC Stamp II has twice the number of I/O pins, twice the execution speed, 5 times the memory (code) space, a serial port and 5 times the resolution on time sensitive commands. In many cases, the BASIC Stamp II fits applications better than the BASIC Stamp I. In some cases, however, this is not true. For example, you may have a need for only a couple of inputs and outputs with which you need to perform relatively simple, non speed-critical tasks. This is a perfect application for the BASIC Stamp I; the BASIC Stamp II would be overkill if used in this way. A slightly different example would be if you have a variable pulse width signal you need to monitor. The BASIC Stamp I can detect and measure a pulse as little as 10 µS wide and as long as 0.65535 seconds wide. The BASIC Stamp II has 5 times the resolution, so it can measure a pulse as little as 2 µS wide, however, only as long as 0.13107 seconds wide. While the better resolution may be helpful, the smaller maximum pulse width measuring capability may prove disadvantageous to your application.Figure 28 depicts the digital integrated circuit/ small-outline integrated circuit (DIP/SOIC) version of the PBASIC2 interpreter chip, since users wishing to construct a BS2 from discrete components are most likely to use those parts. Even if this is not shown on the schematic, it is very wise to place a 100nF capacitor near the power pins (pin 2 and 4) of PIC16C57C IC (with the emended basic stamp 2 interpreter). This will improve the power's supply behavior when you use medium or high frequency pulses.

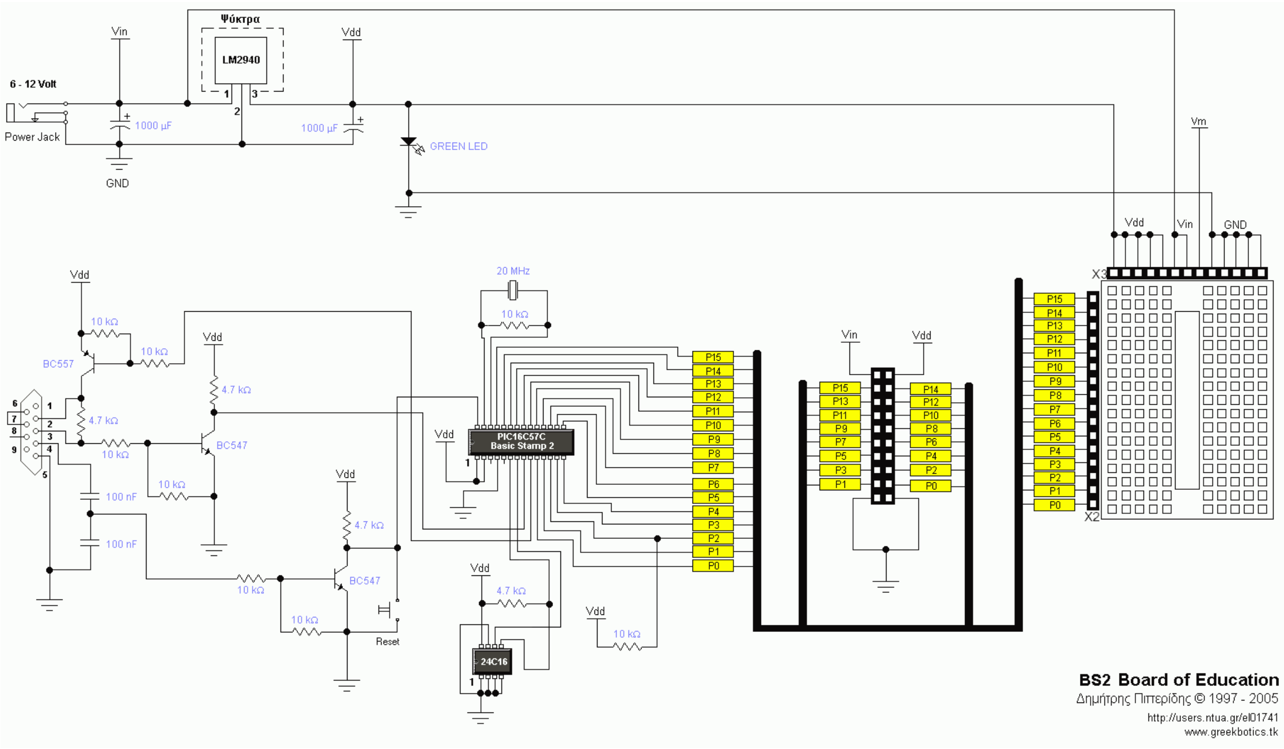


Figure 28. Schematic diagram for BASIC Stamp II.

11. The PBASIC Language

The programming language used to accomplish these tasks is **PBASIC**, which stands for:

- **P**arallax - Created by PARALLAX Inc, a company
- **B**eginners - Made for beginners to learn how to program computers
- **A**ll-purpose - Powerful and useful for solving many different kinds of problems
- **S**ymbolic - Using symbols (terms that resemble English word/phrases)
- **I**nstruction - To tell a computer what to do
- **C**ode - In terms that the computer (and you) can understand

Parallax Inc. created PBASIC which is a micro-controller based version of BASIC. This special language has familiar BASIC instructions such as FOR..NEXT, IF..THEN and GOTO as well as some useful extra instructions that are especially for input and output (I/O). Programs can be written using the STAMP programming software and downloaded on a serial port to the BASIC Stamp.

The simple and basic code created by Parallax Inc. for beginners is :

```
' What's a Microcontroller - Ch01Prj02_ FirstProgramYourTurn.bs2
' BASIC Stamp sends message to Debug Terminal.
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Hello, it's me, your BASIC Stamp!", CR
DEBUG "What's 7 X 11?", CR, "The answer is: ", DEC 7 * 11
END
```

Here,

the first line is a title of the program and third and fourth line states the directive:

a. stamp version

b. PBASIC version

12. WORKING MECHANISM

Distance is a value between the object and the obstacle which is responsible for the robots and automated machinery mechanism, which can be set in a code that will be used in the obstacle detection. Let us take a right IR led and a right servo as reference.

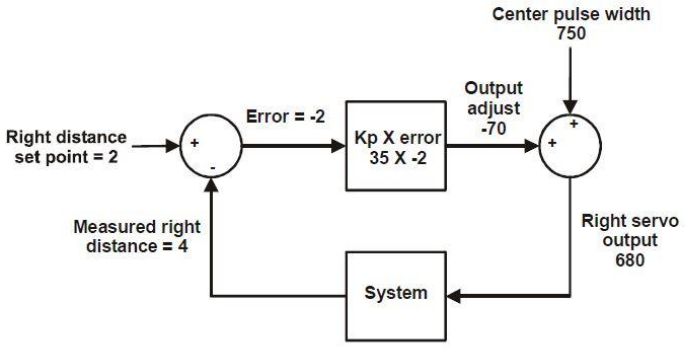


Figure 29. Servo as reference.

Let us assume that the BOE-Bot has maintained a distance of 2 between itself and the obstacle to detect. Here the measure distance between them is 4 meaning that the obstacle is little too far away. So now the error is calculated between the difference of the set point and the measured distance. Kp is a proportional constant whose value is 35. Proportional gain is the amount the error signal is multiplied by directly. Integral gain is the inverse of the time constant applied to the error signal. In a P-I controller, the outputs of the integral and proportional blocks are added to produce the signal that the controller attempts to output. Controllers come in many varieties. The simplest - but certainly not the only one used - is a proportional controller. That's what we will consider here, but there are also integral controllers, and controllers that blend integral, proportional and derivative control and lots of others. In a proportional controller, the control action is proportional to the error, and we can represent the controller as a gain, Kp.

Error = Right distance set point - Measured right distance

= 2-4

Output adjust = Error x Kp

= -2 x 35

= -70

Right servo Output = Output adjust + Center pulse width

= -70 + 750

= 680

Here, 750 is the servo's center pulse with and the result 680 is the right servo's pulse width that will make the robot rotate about ¾ clockwise speed.

In addition, we can achieve that output in the following way:

Right Servo Output = (Right distance set point – Measured right distance) × Kp

+ Center pulse width

Note that the left servo has the same mechanism as the right servo that was explained above but the value of Kp is -35 instead of +35.

Left servo output = (Left distance set point – Measured left distance) Kp+ Center pulse width

= ((2 – 4) x –35) + 750

= 820

The robot will rotate about ¾ anti-clockwise.

The figure 30 shows an example of how the BOE-Bot can test for distance using frequency.

Let's take a example that the object is in zone 3That means that the object can be detected when 37500 and 38250 Hz is transmitted, but it cannot be detected with 39500, 40500, and 41500 Hz. If the object is moved into Zone 2, then the object can be detected when 37500, 38250, and 39500 Hz are transmitted, but not when 40500 and 41500 Hz are transmitted. Meaning that object can be detected on transmission of 37500 and 38250 Hz, but not with 39500, 40500, and 41500 Hz. If the object is moved into Zone 2, then the object can be detected when 37500, 38250, and 39500 Hz are transmitted. In order to test the IR detector at each frequency, we need to use FREQOUT to send five different frequencies and test at each frequency to find out whether the IR detector could detect the object.

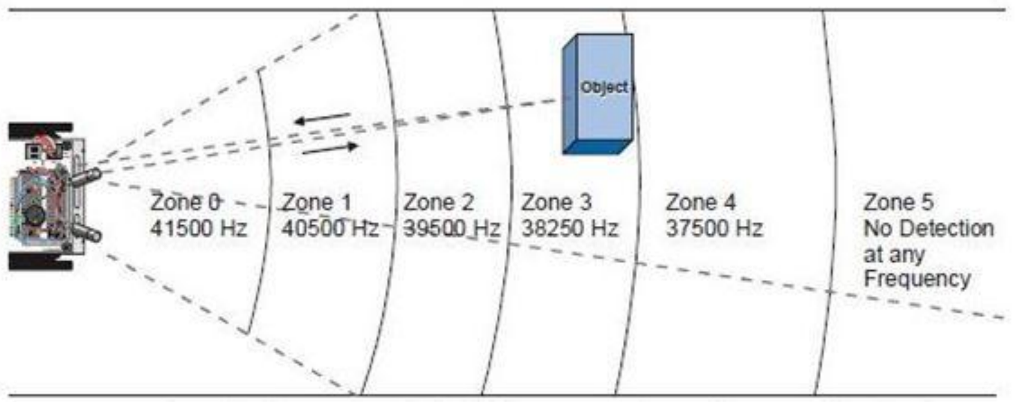


Figure 30. Mechanism of the detection of obstacle by IR Sensors.

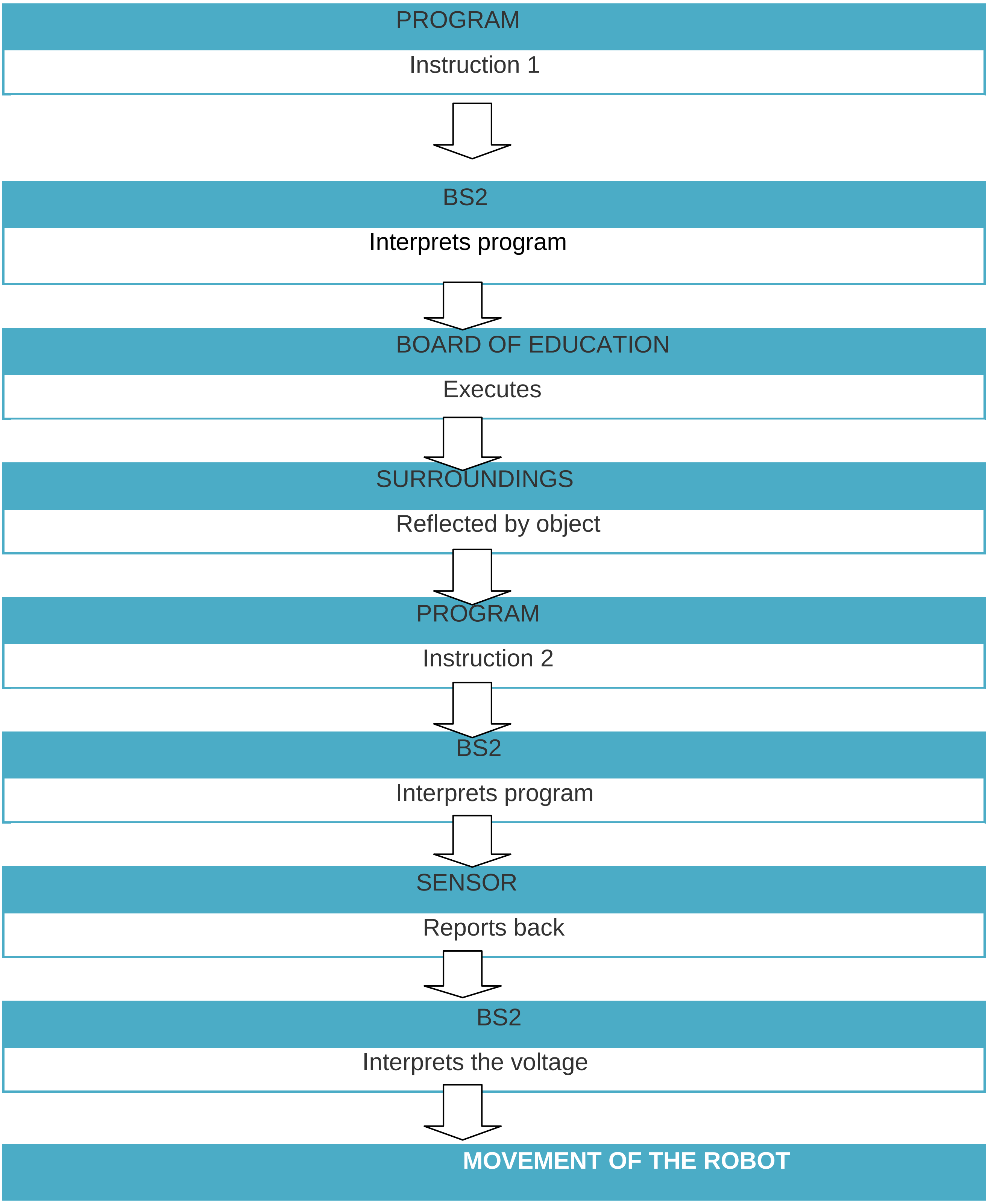


Figure 31. HOW the BOE-Bot works.

As shown in the display in the scheme above, BOE-Bot works in a specific order. After the instruction to emit the signal, it will transmit signal as shown in the figure below. If the IR sensor mounted on the BOE-Bot picks it up, it will vary an internal variable resistor and return 6 voltage back to the BS2.Then; BS2 will give order to turn the other way if it detects the object through the returned signal. There are always many IR rays emitted and received by emitter and receiver respectively, but depending on the position of the object, in reference to the robot, either of the two used sensors receive more deflected rays and thus giving instruction to the robot which side is object detected. Then if the left sensor receive more IR rays, it will move towards right and vice versa.

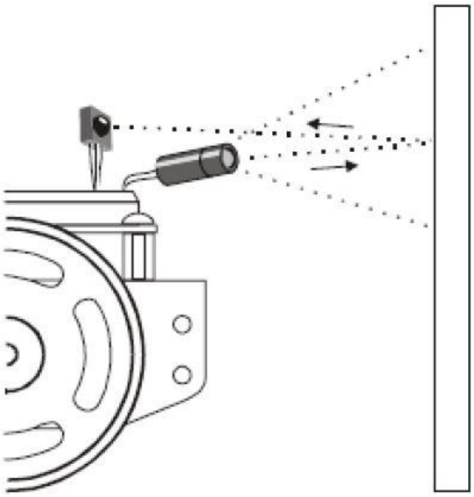


Figure 32. IR Sensor (left) & IR LED (right).

13. RESULT

The outcome of the thesis is a simple, PC-controlled robot controlled which moves around detecting the obstacles on its way and thus making its way through the free space avoiding all the obstacles it encounters. After the completion of the robot, the robot can now emit IR ray through the IR emitter, and the IR receiver will detect receive the bounced IR ray that strikes the obstacles if there are any. If there are any obstacles and IR collides with them, then brain of the robot (micro controller) will calculate the distance.

The robot is made activated after the switch on the activity board is turned on. If there seems to have any obstacle with in the distance set to it, it will move to the opposite direction of the obstacle detected. That is, if the right sensor detects the object on its way, it will turn left and start moving straight again and if left sensor detects an obstacle, it will turn right and continue moving. Additionally, if there seems to be obstacle all around the robot within the detection distance, robot will keep on rotating 360°.

Overall, the BOE-Bot robot performed very well. It had only minor problems with the left sensor which was later on corrected by changing the pin on the breadboard. However, there had to be many adjustments made to robot in order to achieve our destination before the tests started.

After the project, students can design alternative technical solutions based on existing requirement specifications. They can independently and proactively analyze and solve complex technical challenges, as well as to study and evaluate the solution proposals of others which will develop knowledge, skills and attitudes. The students will be familiar with the most important technological and mathematical tools. Following main competences will be developed after the completion of the project :

- Mathematical and Scientific Skills
- Hardware Expertise
- Software Expertise
- Embedded Software: Proficiency in SW Engineering in Embedded Systems
- Development competence
- Electro technical and other basic competence
- Competence in machines and systems
- Individual scale innovation competences

14. CONCLUSION

Today we are in the world of robotics. Knowingly or unknowingly, we have been using different types of robots in our daily life. The aim of the thesis is to evaluate what students can learn about the fields of engineering, mechatronics, and software development as they design, construct, and program an autonomous robot. This will to provide a guideline to the students who are new in the world of Arduino and help them to understand about embedded system, IR sensors, microcontroller and how to make a robot using Arduino.

This project had the goal of manipulating different parts of the robot to make it react according to our desire. The overall end objective was to make the robot follow a black track just by programming it to use the sensors attached to it. The aim of the thesis is to provide simple guidelines to the new students and beginners who are interested in this type of project. Although the outcome was simple, as mentioned earlier, the project makes the new students familiar with Arduino, the working mechanism of it and future aspects of it in a simple and understandable way. Although the thesis project is very little about the robot's use in real world, with the help of guidelines and the abundance of resources the outcome, it could be very beneficial for many people and different sectors of the world depending on the sensors and features required as per necessity.

The goal of this project is to get students interested in and excited about the fields of engineering, mechatronics, and software development as they design, construct, and program an autonomous robot. The thesis not only gives detailed information about Arduino and the use of App Inventor for android application design but also about the IR, IR sensors, PBASIC programming platform to the new students or beginners. The guidelines provided are very simple to use and understand thus, making it very easy for the new students to build a foundation in their Robotics learning.

This project is very much helpful to the first year students who have a keen desire and interest in the robotics and specially Arduino robotics. From this students will learn about servo motors and how to program them. They will also program a specific robot, the BOE-Bot. The BOE-Bot uses the Board of Education as its brain. Students will learn how to turn the BOE-Bot into a fully autonomous unit. Even being a powerful little device, microcontroller would be nothing without its hardware. It's just like a brain that operates a body whereas in this case, the body is made up of many components, including motors, lights, and sensors. The motors provide locomotion, the sensors act as triggers for the motors, and the lights are used primarily as warning devices.

The project gives idea to the new students on different criteria of Understandings, Knowledge and skill.

Understandings:

1. Microcontrollers are used to control many everyday products like robots, garage door openers, traffic lights, and home thermostats.
2. A servo motor is one that delivers continuous motion at various speeds.
3. Microcontrollers can be programmed to sense and respond to outside stimuli.
4. servo motor and what parameters does it use in programming code

Knowledge and skills:

Through this project, student will gain basic knowledge and skills regarding servo, program and mathematics to calculate program values. It is expected that students will:

1. Program a servo motor.
2. Program and test an autonomous robot.
3. Use mathematics to calculate programming values.

Students will learn how to:

1. Program the BOE-Bot to perform basic maneuvers (i.e. forward, backward, left, right, spin)
2. Calculate the number of pulses required to make the BOE-Bot travel a predefined distance.
3. Program the BOE-Bot to gradually accelerate into and decelerate out of a maneuver.
4. Write subroutines to perform basic maneuvers.
5. Record complex maneuvers into the BOE’s memory.

REFERENCES

2R Hardware & Electronics: Arduino Uno. 2013. 2R Hardware & Electronics: Arduino Uno. [ONLINE] Available at: http://2r-he.blogspot.fi/2011/04/arduino-uno.html. [Accessed 16 May 2013].

Application-Specific Integrated Circuits (ASIC's). 2013. Application-Specific Integrated Circuits (ASIC's). [ONLINE] Available at: http://www.siliconfareast.com/asic.htm. [Accessed 19 July 2013].

Arduino - Software . 2013. Arduino - Software . [ONLINE] Available at: http://arduino.cc/en/Main/Software. [Accessed 16 May 2013].

Arduino - HomePage . 2013. Arduino - HomePage . [ONLINE] Available at: http://arduino.cc/. [Accessed 16 May 2013].

Arrick Robotics - Comparing A Robot and BOE-Bot . 2013. Arrick Robotics - Comparing ARobot and BOE-Bot . [ONLINE] Available at: http://www.arrickrobotics.com/arobot/arobotboebot.html. [Accessed 15 October 2013].

Banzi, Massimo, 2009. Getting Started with Arduino. 1st ed. Farnham: O'Reilly Media.

BS2 programming board (board of education). 2014. BS2 programming board (board of education). [ONLINE] Available at: http://users.ntua.gr/dpiperid/MyWebPage/Contructions/Bs2BoardEN.htm. [Accessed 20 January 2014].

Embedded Systems Design – Heath Steve - Google-kirjat. 2013. Embedded Systems Design - Steve Heath - Google-kirjat. [ONLINE] Available at: http://books.google.fi/books?id=BjNZXwH7HlkC&pg=PA2&redir_esc=y#v=onepage&q&f=true. [Accessed 18 July 2013].

Free Merriam-Webster Dictionary. 2013. y. [ONLINE] Available at: http://www.merriam-webster.com/dictionary/robot. [Accessed 18 August 2013].

History of Computers and Computing, Birth of the modern computer, The bases of digital computers, Integrated Circuit. 2014. History of Computers and Computing, Birth of the modern computer, The bases of digital computers, Integrated Circuit. [ONLINE] Available at: http://history-computer.com/ModernComputer/Basis/IC.html. [Accessed 31 January 2014].

Infra Red Light. 2013. Infra Red Light. [ONLINE] Available at: http://astro.uchicago.edu/cara/about_cara/defn/irlight.html. [Accessed 18 November 2013].

Infrared-based obstacle avoider- AIEPIC Project 2009-2010. 2013. Infrared based obstacle avoider- AIEPIC Project 2009-2010. [ONLINE] Available at: http://www.siliconindia.com/aiepic/project/infrared_based_obstacle_avoider-pid=8419.html. [Accessed 10 November 2013].

Introduction To Embedded Systems. 2013. Introduction To Embedded Syestes. [online] Available at: http://www.slideshare.net/yayavaram/introduction-to-embedded-systems-2614825. [Accessed 21 July 2013].

Obstacle Avoiding Robot Report Robot23. 2013. Obstacle Avoiding Robot Report Robot23. [ONLINE] Available at: http://www.slideshare.net/abhi230789/obstacle-avoiding-robot-report-robot23. [Accessed 20 November 2013].

Obstacle Detctor Robot. 2013. Obstacle Detctor Robot. [ONLINE] Available at: http://www.slideshare.net/NikitaKaushal/obstacle-detctor-robot. [Accessed 21 July 2013].

Obstacle Detection and avoidence Robot (1) Computer Science Project Ideas. 2013. Obstacle Detection and avoidence Robot (1) Computer Science Project Ideas. [ONLINE] Available at: http://seminarprojects.com/Thread-obstacle-detection-and-avoidance-robot#ixzz2lvDgFg7F. [Accessed 10 November 2013].

Oxford dictionary (British & World English). 2013. [ONLINE] Available at: http://www.oxforddictionaries.com/definition/english/robot. [Accessed 18 August 2013].

The Making of Arduino - IEEE Spectrum. 2013. The Making of Arduino - IEEE Spectrum. [ONLINE] Available at: http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino. [Accessed 16 May 2013].

Webopedia Computer Dictionary. 2013.. [ONLINE] Available at: http://www.webopedia.com/TERM/I/integrated_circuit_IC.html. [Accessed 3 August 2013].

Webopedia Computer Dictionary. 2014.. [ONLINE] Available at: http://www.webopedia.com/TERM/M/microcontroller.html. [Accessed 20 January 2014].

Wheat, Dale, 2011. Arduino Internals. 1st ed. New York: Apress.

Software Download link:

1. BASIC Stamp Editor Software

BASIC Stamp Editor Software | Parallax Inc. 2014. BASIC Stamp Editor Software | Parallax Inc. [ONLINE] Available at: http://parallax.com/downloads/basic-stamp-editor-software. [Accessed 15 January 2014].

2. Parallax USB Driver Installer:

Parallax USB Driver Installer | Parallax Inc. 2014. Parallax USB Driver Installer | Parallax Inc. [ONLINE] Available at: http://parallax.com/downloads/parallax-usb-driver-installer. [Accessed 15 January 2014].

APPENDIX:

```
' -----[ Title ]-----------------------------------------------------
' Robotics with the Boe-Bot ObstacleAvoidingRobot.bs2
' Navigate left/right, 3-way, and 4-way intersections.
' {$STAMP BS2}                              ' Stamp directive.
' {$PBASIC 2.5}                             ' PBASIC directive.
DEBUG "Program Running!"
 ' -------------[ Constants ]------------------------------------------------
Kpl                     CON          35           ' Left proportional constant
Kpr                    CON           -35           ' Right proportional constant
SetPoint               CON           3            ' 0-1 is White, 4-5 is Black
CenterPulse             CON          750
Turn90Degree           CON          30            ' Pulses needed for 90 turn
RightLED               PIN           1             ' LED Indicators
LeftLED                PIN           10
' -------------------[ Variables ]------------------------------------------------
freqSelect             VAR          Nib            ' Sweep through 5 frequencies
irFrequency            VAR          Word           ' Freq sent to IR emitter
irDetectLeft            VAR          Bit          ' Store results from detectors
irDetectRight          VAR          Bit
distanceLeft           VAR          Nib           ' Calculate distance zones
distanceRight          VAR          Nib
pulseLeft              VAR          Word           ' Servo pulseWidths
pulseRight             VAR          Word
numPulses              VAR          Byte           ' Count total pulses
fwdPulses              VAR          Byte           ' Count forward pulses
counter                VAR          Byte
```

```
isStuck                         VAR             Bit             ' Boolean variable,is bot stuck?


' ---------------------[ Initialization ]-------------------------------------------
 FREQOUT   4,  2000,  3000


' --------------------[ Main Routine ]---------------------------------------------


DO
GOSUB          Get_Ir_Distances                                 ' Read IR sensors
 GOSUB          Update_LEDs                                     ' Indicate white/black line


' Calculate proportional output and move accordingly.
 pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
 pulseRight = SetPoint - distanceRight * Kpr + CenterPulse
 GOSUB Send_Pulse
 GOSUB          Check_For_Intersection                          ' Are we stuck at intersection?
 IF (isStuck = 1)                THEN
                GOSUB           Make_Noise                      ' Audible indication
                GOSUB           Navigate_Intersection           ' Navigate through it
 ENDIF
 LOOP
' ------------------------[ Subroutines ]-----------------------------------------------


Navigate_Intersection:
' Go forward until both sensors read white, through the intersection.
 DO
                 pulseLeft = 850:                pulseRight = 650 ' Forward
```

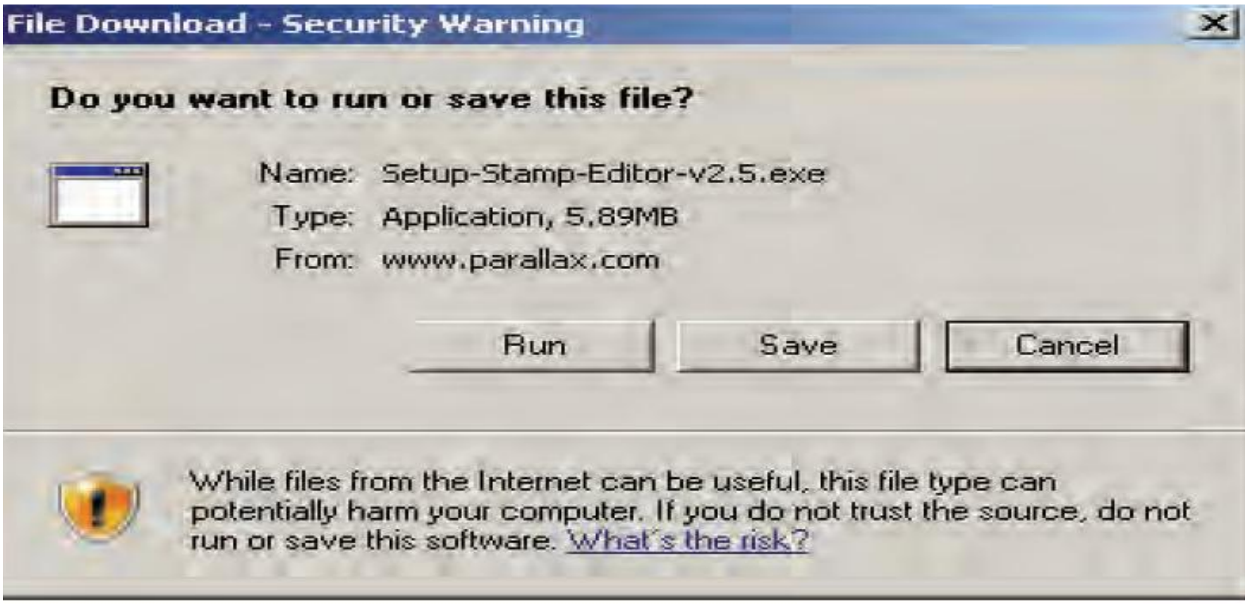
```
                    GOSUB          Send_Pulse

                    GOSUB          Get_Ir_Distances

                    GOSUB         Update_LEDs

LOOP UNTIL (distanceLeft <=2) AND (distanceRight <=2)

GOSUB        Stop_Quickly                              ' Don't coast forward


' Now back up until one detector sees the black.L & R turn will see

' black on one detector.3- or 4-way will see both black, turn toward

' whichever the bot sees first (random).

DO

                    pulseLeft = 650:                    pulseRight = 850 ' Backward

                     GOSUB         Send_Pulse

                     GOSUB         Get_Ir_Distances

                    GOSUB          Update_LEDs

LOOP UNTIL (distanceLeft >=4) OR (distanceRight >=4)

GOSUB         Stop_Quickly                                                 ' Don't coast backward


' Make 90 degree turn in direction of the detector which sees black

                     IF (distanceLeft >=4)          THEN                         ' Left detector reads black

                     FOR counter = 1                TO Turn90Degree             ' Turn 90 degrees left

                    PULSOUT        13, 750                                     ' without proportional control

                    PULSOUT        12, 650

                    PAUSE          20                                          ' so use PAUSE 20

NEXT

ELSEIF (distanceRight >=4) THEN                                                ' Right detector reads black

FOR counter = 1 TO Turn90Degree                                                 ' Turn 90 degrees right

PULSOUT       13, 850
PULSOUT      12, 750

PAUSE        20

NEXT

ENDIF


' At this point the Boe-Bot should have turned 90 degrees

' to follow the intersection. Continue following the black line.


RETURN


Check_For_Intersection:

' Keep track of no. of pulses vs the forward pulses. If there are less

' than 30 forward pulses per total of 60 pulses, robot is likely stuck

' at an intersection.


            isStuck = 0                                   ' Initialize Boolean variable

            numPulses = numPulses + 1                      ' Count total pulses sent


           SELECT      numPulses

                        CASE < 60

                        IF (pulseLeft > CenterPulse) THEN

                        fwdPulses = fwdPulses + 1         ' Count forward pulses

                       ENDIF                               ' (forward is any pulse > 750)


                       CASE = 60                          ' If we have sent 60 pulses

                       IF (fwdPulses < 30) THEN           ' how many were forward?

                        isStuck = 1                       ' If < 30, robot is stuck
```

```
                ENDIF


                CASE > 60
                 numPulses = 0                 ' Reset counters back to zero
                fwdPulses = 0                  ' (Could reset in =60 case but
                ENDSELECT                      ' it spoils cool Make_Noise)
        RETURN


Make_Noise:
' Makes an increasing tone, proportional to number of forward pulses
 FOR counter = 1 TO fwdPulses STEP 3
 FREQOUT 4, 100, 3800 + (counter * 10)
 NEXT
 RETURN
```

INSTALLATION OF THE SOFTWARE

The BASIC Stamp editor software was used in the project and needed to be installed in a PC as the robot to be assembled and coded is to be controlled through the PC. The first step was to go to the Parallax web site and download the software. BASIC Stamp editor v2.5.3 was downloaded from the site of Parallax Inc.

After the file had been downloaded, we clicked on "save" the file and then the prompts were followed. After the completion of the download, run prompt was clicked to verify the installation of the BASIC Stamp editor software.



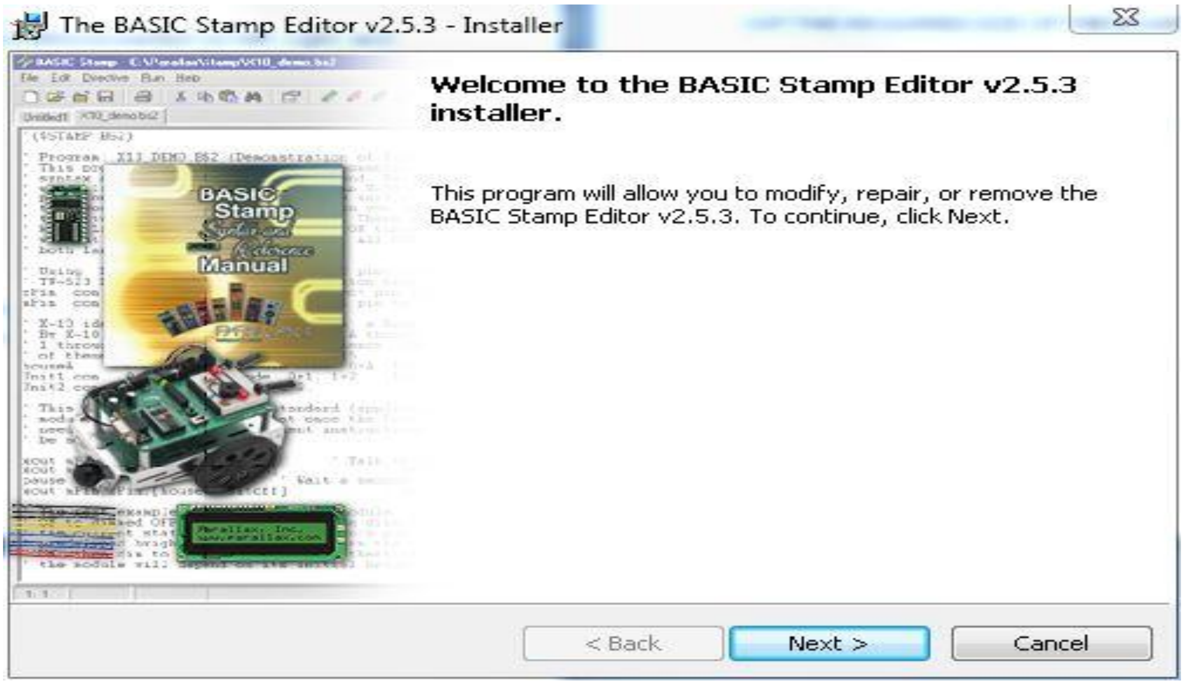
File Download - Security Warning

Do you want to run or save this file?

Name: Setup-Stamp-Editor-v2.5.exe
Type: Application, 5.89MB
From: www.parallax.com

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

Figure 33. File download window.



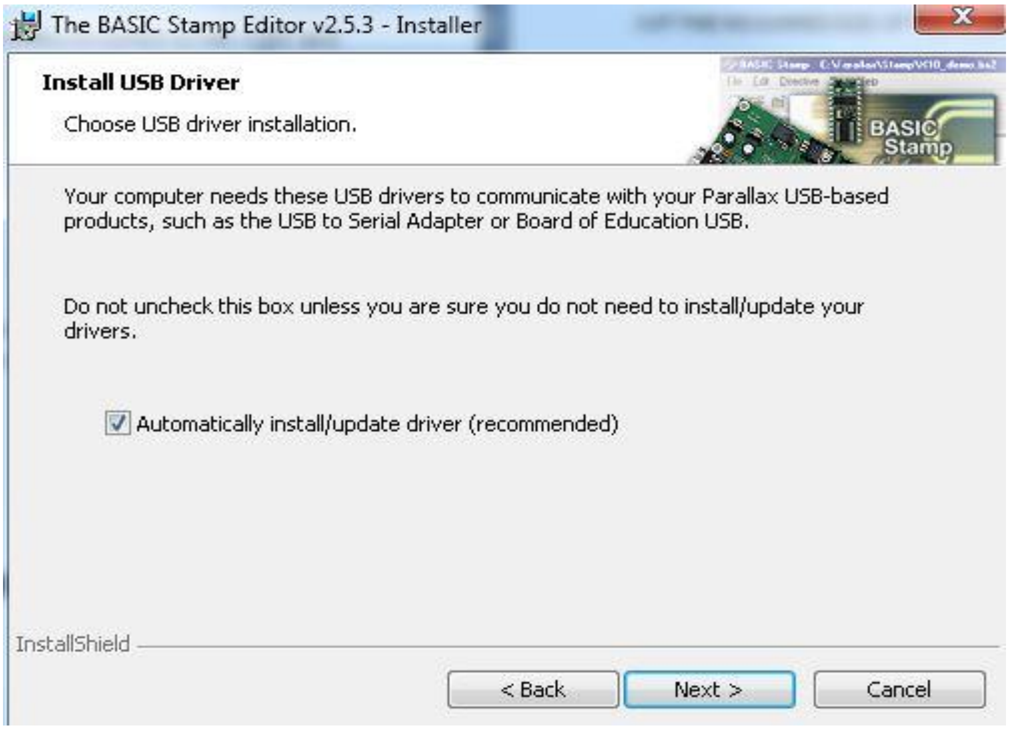
The BASIC Stamp Editor v2.5.3 - Installer

Welcome to the BASIC Stamp Editor v2.5.3 installer.

This program will allow you to modify, repair, or remove the BASIC Stamp Editor v2.5.3. To continue, click Next.

Figure 34. BASIC Stamp Editor Installer Window.

When the “Install USB Driver” message appeared, the checkmark box was left checked in order to automatically install/update the driver and next was clicked.



Install USB Driver

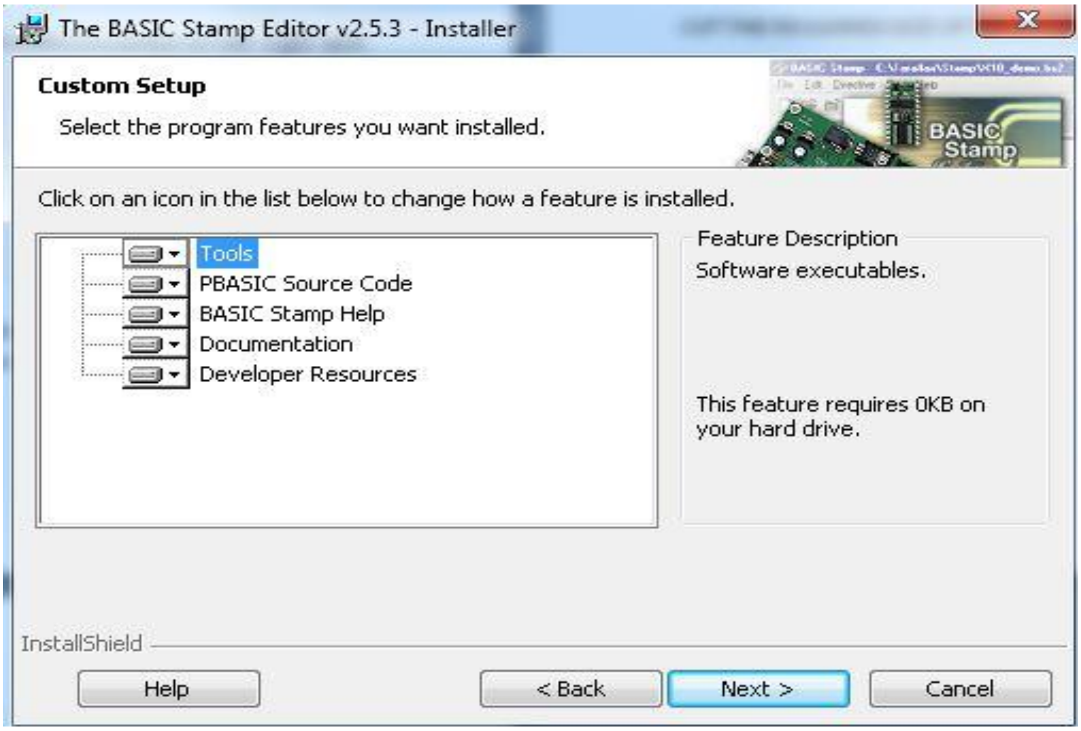
Choose USB driver installation.

Your computer needs these USB drivers to communicate with your Parallax USB-based products, such as the USB to Serial Adapter or Board of Education USB.

Do not uncheck this box unless you are sure you do not need to install/update your drivers.

☑ Automatically install/update driver (recommended)

Figure *35. Install USB Driver Message Window.*



Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

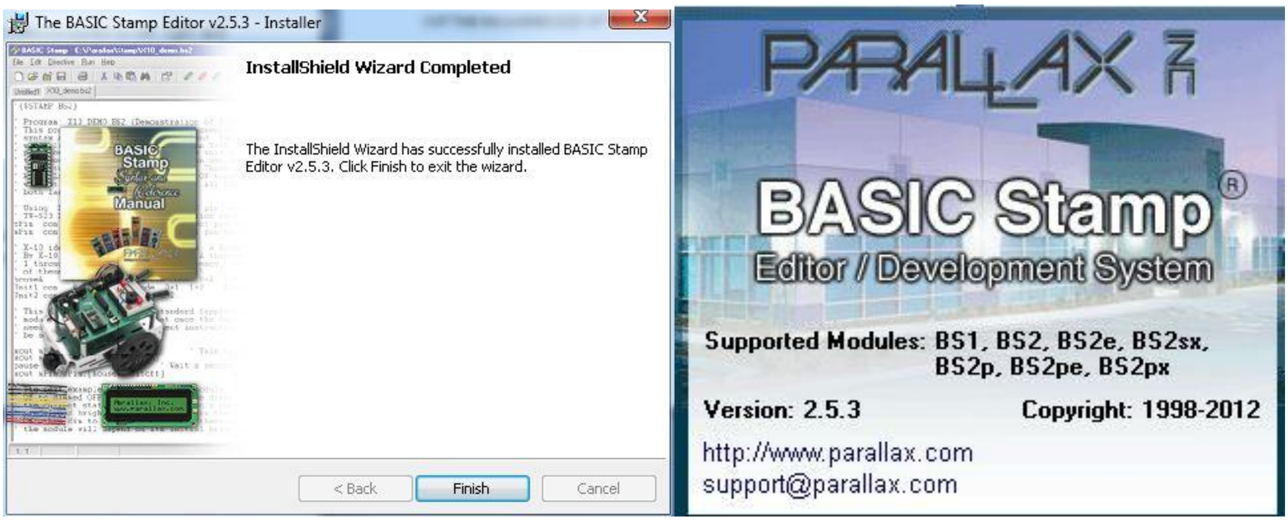
Feature Description

Software executables.

This feature requires 0KB on your hard drive.

Figure 36. Custom setup window (click next).

When the “Ready to Install the Program” message appeared, the “install” button was clicked. After the message appeared that installation was finished, "finish" was clicked.


The BASIC Stamp Editor v2.5.3 - Installer
< Back
Finish
Cancel
PARALLAX
BASIC Stamp®
Editor / Development System

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed BASIC Stamp Editor v2.5.3. Click Finish to exit the wizard.

Supported Modules: BS1, BS2, BS2e, BS2sx, BS2p, BS2pe, BS2px

Version: 2.5.3

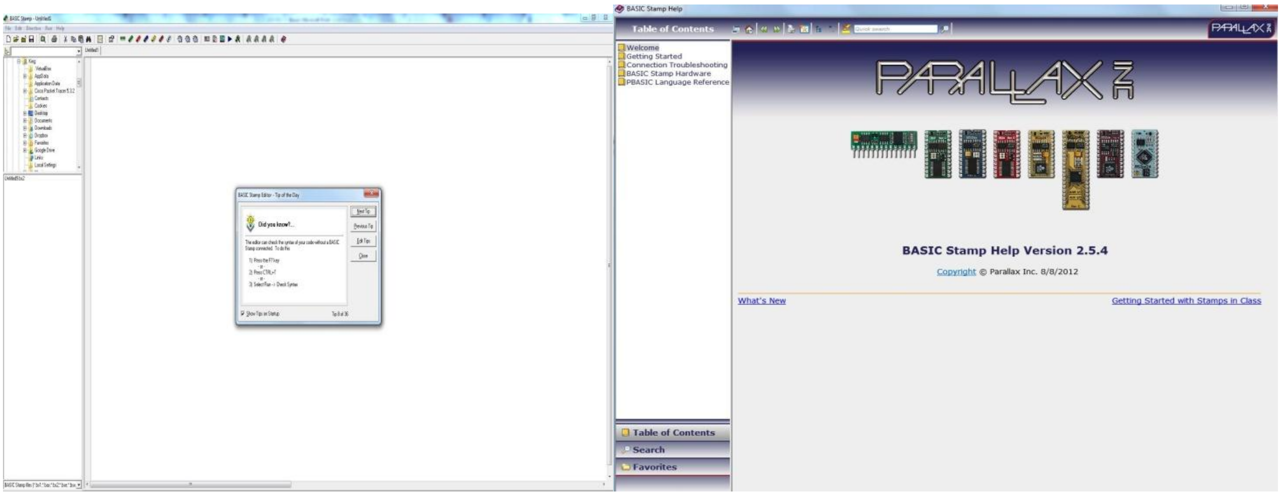
Copyright: 1998-2012

http://www.parallax.com

support@parallax.com

Figure 37. BASIC Stamp Editor Installation Completed.

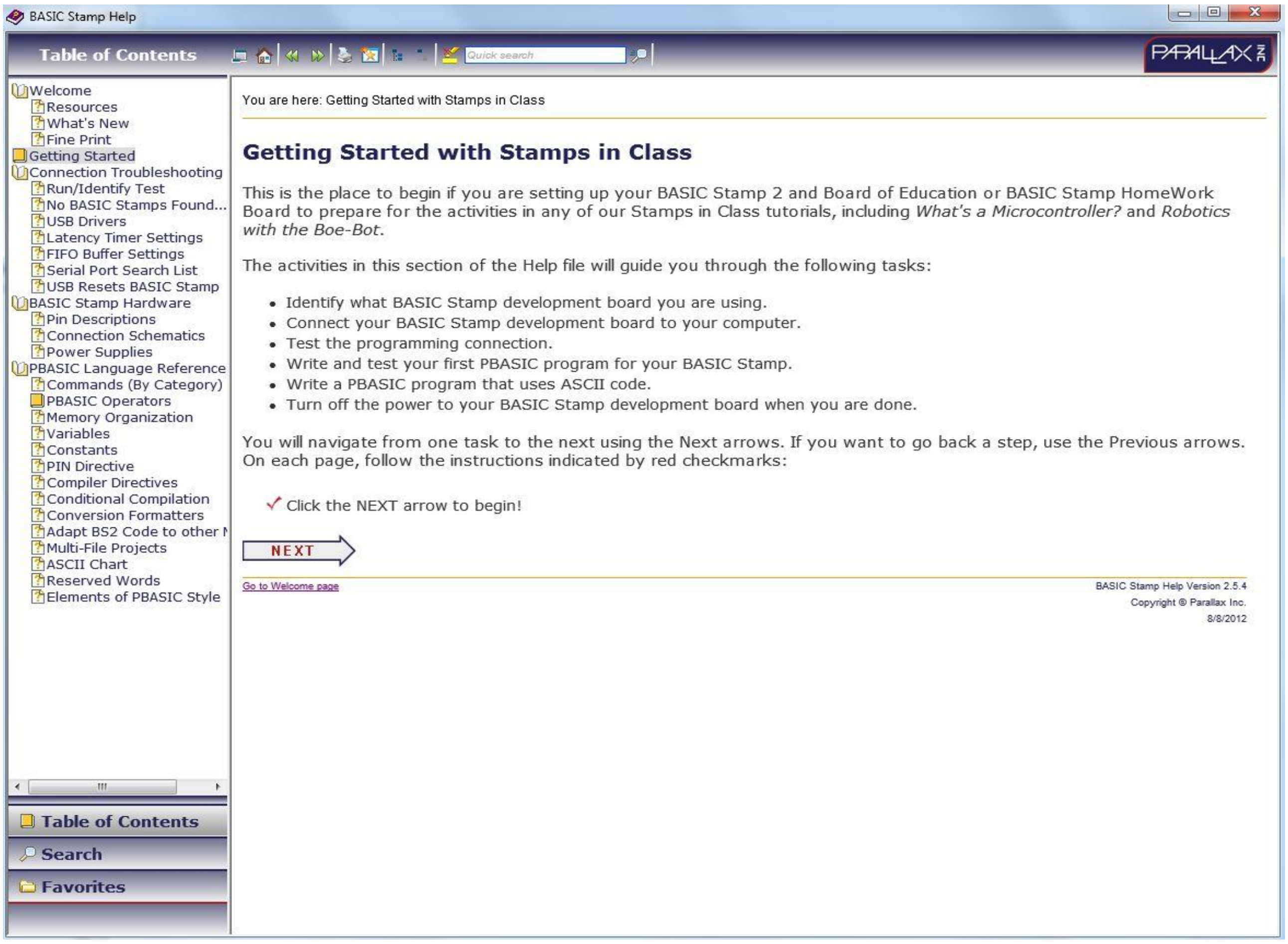
After the installation of the downloaded file of BASIC Stamp Editor, it is always useful and easier to go through the *help* directory when different ideas and information are available.



BASIC Stamp Help Version 2.5.4

Copyright © Parallax Inc. 8/8/2012

Figure 38. : BASIC Stamp Editor.



You are here: Getting Started with Stamps in Class

Getting Started with Stamps in Class

This is the place to begin if you are setting up your BASIC Stamp 2 and Board of Education or BASIC Stamp HomeWork Board to prepare for the activities in any of our Stamps in Class tutorials, including *What's a Microcontroller?* and *Robotics with the Boe-Bot*.

The activities in this section of the Help file will guide you through the following tasks:

- Identify what BASIC Stamp development board you are using.
- Connect your BASIC Stamp development board to your computer.
- Test the programming connection.
- Write and test your first PBASIC program for your BASIC Stamp.
- Write a PBASIC program that uses ASCII code.
- Turn off the power to your BASIC Stamp development board when you are done.

You will navigate from one task to the next using the Next arrows. If you want to go back a step, use the Previous arrows. On each page, follow the instructions indicated by red checkmarks:

✓ Click the NEXT arrow to begin!

Go to Welcome page

BASIC Stamp Help Version 2.5.4
Copyright © Parallax Inc.
8/8/2012

Figure 39. BASIC Stamp Editor help directory.

The BASIC STAMP Help file gives instructions how to perform actions, for example,:

1. To identify the BASIC Stamp development file being used

2. To connecting the activity board to the computer

3. To troubleshooting programming, if required

4. To write PBASIC program for BASIC Stamp

5. The hardware should be powered down when everything is completed