## Index

### Chapter One: What is Ebot System? ________ 7
- What is the Ebot? ____________________________ 8
- Ebot-8 Pinout diagram ________________________ 9
- Ebot-4 Pinout diagram _________________________ 10
- What is Ebot Blockly? _________________________ 11
- Ebot Blockly User Interface ____________________ 12
- How to Read the Properties of Blocks ___________ 13
- Downloading the Ebot Blockly Program __________ 14
- Connecting Components to the Ebot ______________ 14
- How to use the Guide _________________________ 15

### Chapter Two: Projects ________________________ 17
- Meet the Outputs ___________________________ 18
- Meet the Inputs _____________________________ 19
- Controlling Multiple Elements ________________ 20
- Controlling the Servo Motor ________________ 21
- Controlling the Servo Vocally ________________ 22
- Controlling a DC Motor Using Light ____________ 23
- Controlling a DC Motor Using Infrared (IR) ______ 24
- Technique of the Line Following Robot __________ 25
- Build a Line Follower Robot _________________ 26
- Controlling Motors Using Bluetooth ____________ 27
- Learn to Use the LCD ______________________ 28
- Controlling RGB ____________________________ 29
- Using the Ultrasonic Sensor __________________ 30
- Using the Repeat __________________________ 31
- Using the Potentiometer _____________________ 32
- Controlling LEDs Using the Potentiometer _______ 33
- Controlling Outputs with Custom Code __________ 34
- Using the Temperature Sensor ________________ 35
- Create a digital Thermometer _________________ 36
- Creating a Color Reader _______________________ 37
- Playing Multimedia __________________________ 38
- Using the BreadBoard ________________________ 39
- Using the IF statement _____________________ 40
- Program using C++ Programming Language _______ 41

### Chapter Three: Extra Features ______________ 43
- Connecting the Ebot with your Device _________ 44
- Setting Up Ebot for Bluetooth ________________ 44
- Input Reading Feature ______________________ 45
- Custom input _____________________________ 45
- Live Control ______________________________ 46
- Data Lab _________________________________ 47
- What is Arduino? __________________________ 48
- C++ Programming Language ________________ 48
- Ohm’s Law ________________________________ 49

### Chapter Four: Obstacle Avoiding Car ________ 51
- Plastic pieces used in the project _____________ 52
- Obstacle Avoiding Car Project ___________________ 52
- Project programming _________________________ 56
- Enjoy and improve your skills _____________ 57
- Get certified by testing your knowledge ___________ 57
Chapter One: What is Ebot System?
What is the Ebot?

Ebot is a microcontroller unit. A microcontroller is a small digital computer that contains a processor, memory in the form of RAM, and input/output peripherals. Different than a personal computer, a microcontroller is designed to act as a manager for all control systems. Microcontrollers are used in many different products and devices such as car engine controllers, stationary machinery, electrical tools and devices, medical equipment, remote controllers, and many others. They are popular due to low cost and inclusion of everything needed to control different devices as compared to other processing units which use separate processors for inputs and outputs.

The Ebot comes in two different models with different specifications to suit the user’s needs and requirements. Both models contain general inputs, general outputs, and high-current outputs.

1. **General Inputs**: Analog Inputs that allows the user to measure the current passing through it and allows the addition of any external input.
2. **General Outputs**: Outputs that can generate a current up to 25 mA and 5 volts.
3. **High-Current Outputs**: Each of these outputs can generate up to 1 ampere of current and a voltage limited only by the battery’s voltage. This is ideal for outputs that consume a lot of power such as water pumps, air blowers, electrical pumps, etc.

**Ebot-8/ Ebot-8 PRO**
1. 8 general inputs (A0-A7)
2. 8 general outputs (0 - 7)
3. Two motor pins (Left & Right)
4. Bluetooth
5. Real Time Clock (RTC) **PRO**
6. Accelerometer **PRO**
7. Sliding Switch **PRO**
8. 4x RGB LED **PRO**

**Ebot-4**
1. 4 general inputs (A0 - A4)
2. 4 general outputs (0 - 7)
3. 2 motor pins (Left & Right)
4. Accelerometer
5. Gyroscope
6. Temperature Sensor
7. Bluetooth
8. Buzzer
9. 2 x Push Button
10.4 x RGB LED
11. Rechargeable Battery
AMTEL’s AVR family of processors is considered one of the most important electronic units in the world as it is easy to use and provides multiple benefits, most importantly low power consumption and a high level of coordination with each other. These units use C or Assembly as the main programming language, which are two of the major languages in the fields of robotics and circuit design. The Ebot unit was built using the ATMEGA1284P chip from the AVR 8-bit family of processors. This family of processors builds its commands in the form of 8-bits. While the Ebot is based on the ATMEGA1284P chip, a map was drawn, identifying the pins and their counterpart in the ATMEGA1284P chip. This allows an advanced user to program the Ebot for more complex operations.

The schematic has been split into multiple layers to show the links between components. The first layer (displayed in white) shows the pin schematics of the ATMEGA1284P chip. The second layer (displayed in brown) shows the Arduino operation system pins, also known as the bootloader pins. The third and final layer contains the Ebot pins. Notice that in the Ebot layer, the pins have been modified from the original schematic to accommodate the input and output specifications of the Ebot. This helps simplify the programming process for a beginner.
The Ebot-4 is built using the ATMEGA32u4 chip from the AVR 8-bit family of microcontrollers. These chips build commands in the form of 8 bits. The board was designed to simplify the programming process. Since the Ebot-4 uses the ATMEGA32u4 chip, the mapping is drawn to show the pins of the Ebot-4 and the corresponding pins of the ATMEGA32u4. Note that many of the pins are connected internally to components within the Ebot-4, such as RGB LEDs, Buzzers, and Buttons. The advanced user can use this mapping to perform complex programming operations and for use in device manufacturing.

The schematic is divided into multiple layers to show the connection between the different mappings. The first layer (displayed in yellow) shows the pin layout of the ATMEGA32u4 chip. The second layer (displayed in white) shows the pins layout of the Arduino system (Leonardo PIN). The third and final layer contains the schematic pin layout of the Ebot-4.
What is Ebot Blockly

The Ebot Blockly software, developed by Creative Bits Solutions, is a recent programming software used in education.

The goal of Ebot Blockly is to teach programming in a simpler way based on the concept of dragging and dropping program commands, without going into the details of typing complex commands. It also gives the user the option of programming using the Graphical Page or the Code Page. Lastly, the Ebot Blockly adds several features to simplify programming even further, such as Data Lab, Serial Monitor, and Live Control.

The Idea Behind Ebot Blockly

The idea behind Ebot Blockly is to drag blocks from the menus on the left (Input, Output, Flow, Logic, Variable, Advanced, Multimedia) and drop them in the Graphical Page, attaching them to the Start block. Afterwards, any new block can be attached to the other blocks like a magnet.

Implementation Steps:

1. Drag any block from the Input menu and place it under the Start block.
2. To add a block inside another block, drag the block from the left menu and place it inside the previous block as shown.
3. To move a group of blocks, drag the outermost block (the green block in this example) and move it. Note: The program will not work if the blocks are not connected to the Start block (Inactive blocks are shown in gray).
Ebot Blockly User Interface

1. New File
2. Sample Projects
3. Open File
4. Save
5. Save As
6. Print
7. About Company
8. Undo & Redo
9. Settings & Preferences
10. User Interface:
    • Graphic Page: Program using blocks only.
    • Code Page: Program using text input only.
    • Graphic/Code Page: A combination of both.
11. Select Ebot unit (Ebot-4 / Ebot-8)
12. Connection Status (if an Ebot is connected, the unit’s name is shown).
13. Compile: Ensure that the code has no errors.
14. Download the code to the Ebot unit.
15. Code Page: To type and modify block programming using C++.
17. Helping Features (Debug, Serial Monitor, Live Control)
18. Connection Guide depicting how devices should be connected to the Ebot.
19. Blocks in the Graphical Page: More blocks can be added from the left menu; each block’s settings can be changed by clicking on ( ).
20. Center code in Graphical Page
22. Delete blocks from the Graphical Page.
How to Read the Properties of Blocks

Every block in the Ebot Blockly program has several properties, in which the user can control based on the project. To open the properties of a block, the user must clicks on the ◌ after dragging the block to the graphical page. Below is a description of the different blocks and their properties:

**INPUTS**

- **Bluetooth Remote Properties:**
  - Select the button that the block will detect.

- **Color Sensor Properties:**
  - **Range:** Select the range of colors that the sensor will detect, whether it is between two colors or with a variable.

**OUTPUTS**

- **LED & RGB Properties:**
  - **On/Off:** Turn the light on/off.
  - **Color/Variable:** Specify the color of the lighting of the RGB, based on a color or a variable.
  - **Number of LEDs:** The number of LEDs in the RGB Strip.
  - **Color Wipe:** Some effects that the user can choose from.
  - **Delay:** Time for the lighting to take effect.

- **Servo Properties:**
  - **Angle:** The angle of the Servo
  - **Variable:** Move the Servo based on a variable.

- **DC Motor Properties:**
  - **Speed:** The speed of the DC motor’s rotation (0: Stop, 10: Fastest)
  - **Variable:** Control the speed of the DC motor

- **Buzzer Properties:**
  - **Freq:** The frequency of the sound.
  - **Duration:** How long the sound will play:
    - Limited: Duration specified by the user.
    - Forever: Play the sound forever.
    - Variable: Play the sound for a duration equal to the value of the variable.

**Button Properties:**

- **Pressed:** When the button is pressed.
- **Released:** When the button is released.

**Other Input Properties:**

- **Range:** Specify the range of values that the sensor will detect.
  - **Inside:** The range to be detected is between the two values specified.
  - **Outside:** The range to be detected is everything except what is between the two values specified.

**LCD Properties:**

- **Send:** To write the text to be displayed.
- **Row:** The line in which the writing will be displayed. There are two rows (0, 1).
- **Column:** Where the first letter of the text will be written. Can be between 0 and 15.
- **Delay:** The time that the text will stay on the screen.
- **Clear LCD:** Clear the LCD after the delay.
- **Auto Scroll:** Move the text up automatically.

**How to Read the Properties of Blocks**

Every block in the Ebot Blockly program has several properties, in which the user can control based on the project. To open the properties of a block, the user must clicks on the ◌ after dragging the block to the graphical page. Below is a description of the different blocks and their properties:
Connecting components is one of the most important jobs that must be done before starting to program. The user must list the components to be used in the program and categorize them as inputs or outputs. The user needs to determine which pin to connect to which input/output. Keep in mind it is incorrect to connect an input to an output pin and vice versa. The diagram describes how the user can choose any of the 8 pins to connect inputs and outputs.

Warning! An input reads the voltage of what is connected to it and compares it to 5 volts. Therefore, it is important to use sensors that work with 5 volts. Sensors that use higher voltages will not work. Sensors that use less than 5 volts will burn.

All sensors contain three wires (White-Red-Black) that are essential in learning about the movement of current and correct connections to the board. The red (+) wire is the positive of the battery and the black (-) wire is the negative of the battery, both of which are used to power up the sensor. The white (Signal) wire is used to transmit the signal from the sensor to the board. Beware that the black wire is always on the left side of the board.

For Smart Devices:
1. Look for Ebot Blockly in your application store (Google Play, AppStore) or scan the QR code at the bottom of the page.
2. To activate bluetooth, download the computer version and set up your Ebot to use Bluetooth (See page 44).

For Computers:
1. To download the program go to www.Ebots.cc
2. In the Downloads section click the link next to your operating system.
3. Open the downloaded file and follow the Implementation Steps of the installer.

The Ebot Blockly program was developed for all commonly used operating systems on both computers and mobile devices.
How to use the Guide ..

This guide provides a number of simple projects that will allow you to attain an advanced level in using the Ebot. On the next page you will find an index that will show you the different parts of the projects.

1. Project Description: This part contains a description of the project and its most important technologies.
2. Implementation Steps: A step-by-step guide to building the project.
3. Connection Guide: How and where to physically connect the components to the Ebot board.
4. Project Programming: The colored blocks represent the programmable elements which exist within the Ebot Blockly software. These blocks are divided into: Inputs, Outputs, Flow, Logic, Variable, Advanced, Multimedia.
5. General Notes: In this section, topics that are relevant to the project are discussed. It can also contain suggestions on improving the project.
6. Additional information: This part refers to another page with detailed information about specified technology used in the project.
Chapter Two: Projects
Meet the Outputs

**Outputs** are elements that execute orders using a physical manifestation (Movement, Light, Sound).

**Buzzer**: is a mechanical output device that converts electrical energy to sound energy, and uses different frequencies to make different sounds. This device is commonly used in alarms, such as alarm clocks or emergency alarms.

In this project, the user will learn to program the Buzzer to generate an alarm.

Implementation Steps

1. Connect the project as shown.
2. Drag the Buzzer block from the Output menu to the screen and attach it under the Start block.
3. Click the (i) button inside the Buzzer block and choose the settings as shown.
4. Connect the Ebot through USB.
5. Click on the Download button (located on the top right of the screen).

**General Notes:**

The Buzzer can be modified to make different sounds by changing the Frequency and Duration settings.

Try placing multiple Buzzer blocks with different frequencies and durations to make your very own alarm!
You can program the Button to make a different sound when the Button is released by selecting **Released** in the Button’s settings.

**Inputs** are devices that sense the physical environment around us (Light, Touch, Sound, and more) and transform it into information that the Ebot can read. The Ebot then processes this information and sends signals to the outputs to be executed.

**Button**: is a mechanical device that closes the circuit when it is clicked, allowing the Ebot to read this information.

In this project, the user will learn how to use the Button to control the Buzzer.

**Implementation Steps**

1. Connect the project as shown.
2. Drag the Buzzer block from the Output menu to the screen and attach it under the Start block.
3. Click the (i) button inside the Buzzer block and choose the settings as shown.
4. Connect the Ebot through USB.
5. Click on the Download button (located on the top right of the screen).

**General Notes:**

You can program the Button to make a different sound when the Button is released by selecting **Released** in the Button’s settings.

Try making a game that uses this concept.
The order of the blocks matters, as they are based on logic principles. In the above example, the program will turn on the LED and then the Buzzer for 1 second, then it will turn off the light and sound.

Try switching the LED and Buzzer blocks and observe the difference!

Light Emitting Diode (LED): An LED is a light device that emits a light when an electric current is passed through it. LEDs usually emit a single color and are popular because they consume little energy, have a long life, and emit a very strong light. This makes the LED one of the most important elements in modern devices such as TV screens, home lighting, and mobile devices.

There is another type of LED known as RGB. An RGB contains three LEDs with separate colors: Red (R), Green (G), Blue (B). These colors are known as primary colors and can be combined to produce any color. The user can specify different intensities for each color, generating any color in existence. As a result, RGBs are now used in colored screens.

In this project, the user will learn how to use the Button to control the LED and the Buzzer.

Implementation Steps

1. Connect the project as shown.
2. Drag the Button block from the Input menu and the Buzzer and LED from the Output menu and place them on the screen as shown.
3. Click the (b)blocks and change their settings to match the code shown.
4. Connect the Ebot through USB.
5. Click on the Download button (located on the top right of the screen).

General Notes:

The order of the blocks matters, as they are based on logic principles. In the above example, the program will turn on the LED and then the Buzzer for 1 second, then it will turn off the light and sound.

Try switching the LED and Buzzer blocks and observe the difference!
Controlling the Servo Motor

Servo Motor: is a motor which has the ability to move to any angle between -90 and +90 degrees, 180 degrees in total. Servo Motors have many industrial uses, such as: robotic arms, moving airplane fins, medical instrumentations, and control systems that require accuracy in movement.

Delay: A Delay is an order that causes the Ebot to wait for a specified time before continuing with the execution of the code. Delay time is specified by the user and can be as short as a microsecond or as long as an entire day.

In this project, the user will learn how to control the Servo Motor using a Button to move it between two different angles.

Implementation Steps

1. Connect the project as shown.
2. Drag the Button from the Input menu and the Servo Motor from the Output menu and connect them as shown.
3. Click the ( ) blocks and change their settings to match the code shown.
4. Connect the Ebot through USB.
5. Click on the Download button (located on the top right of the screen).

General Notes:

Delays play a big role in organizing how the code works, because the Ebot processes orders and a very fast speed. Removing the Delay block will cause orders to be executed so fast your eyes won’t be able to keep up!

Remove the Delay block and observe the problem.
The sound sensor is affected by the sounds existing in its environment. Strong wind speeds can affect it and mistake it for a signal. Therefore, it is important to use different insulators to minimize the noise surrounding the sound sensor. One example of a filter is the Windscreen Filter.

Create a program to set the noise level in the environment and generate a sound alarm when it exceeds a specific limit.

Sound Sensor (Microphone): is a device that transforms sound waves to electrical signals that the Ebot reads as fluctuations in electrical current. The strength of the sound (Loudness) is measured in Decibels (dB), while its frequency is measured in Hertz (Hz).

In this project, the user will learn how to control the Servo Motor using the Sound Sensor to move it to two different angles.

1. Connect the project as shown.
2. Drag the Sound Sensor from the Input menu and the Servo Motor from the Output menu and connect them as shown.
3. Click the (?) blocks and change their settings to match the code shown.
4. Connect the Ebot through USB.
5. Click on the Download button (located on the top right of the screen).

Create a program to set the noise level in the environment and generate a sound alarm when it exceeds a specific limit.
Controlling a DC Motor Using Light

DC Motor: is a device that converts electrical energy into mechanical energy in a rotational manner. DC Motors are used in several household devices, such as hairdryers, sewing machines, electric brooms, etc.

Light Dependent Resistor (LDR): also known as Light Sensor, allows for the detection of light, where the resistance of the device changes based on the intensity of light shining on it. As the intensity of the light increases, the resistance of the sensor decreases, allowing for more electric current to pass through.

In this project, the user will learn how to measure the intensity of light shining on the sensor, using the Input Reading feature. These values will then be used to move the DC motor when there is enough light, and stopping the DC motor when it’s dark.

Implementation Steps

1. Connect the project as shown.
2. Drag the Light Sensor from the Input menu and the DC Motor from the Output menu and connect them as shown.
3. Click the (?) blocks and change their settings to match the code shown.
4. Connect the Ebot through USB.
5. Use the Input Reading feature and use it to change the values in the Light Sensor settings.
6. Click on the Download button.

Notice the difference in the numbers in the first and second Light Sensor blocks. The is used to indicate that the value read is inside the two values specified, while the is used to indicate that the value read is outside the values specified.

Look at Page 45 to learn how to use the Input Reading feature.
Infrared (IR) are waves of electromagnetic energy that are transferred through light. We feel IR energy anytime we feel heat on our bodies. An IR Sensor consists of two components, an emitter and a receiver. The emitter generates IR waves while the receiver detects IR waves bounced back on different surfaces.

In this project, the user will learn how to use the IR sensor and read its value when using a white surface versus a black surface. Using these values, the user will control the DC motor.

**Implementation Steps**

1. Connect the project as shown.
2. Drag the IR Sensor from the Input menu and the DC Motor from the Output menu and connect them as shown.
3. Click the () blocks and change their settings to match the code shown.
4. Connect the Ebot through USB.
5. Use the Input Reading feature and use it to change the values in the Light Sensor settings.
6. Click on the Download button.

**General Notes:**

The IR sensor may not be able to sense black colors drawn using normal colors or printer ink. Therefore, it is recommended to use Permanent Markers for black colors.

Try using different substances for black colors and find the best one!
Line Follower is a technique that uses a number of sensors to make the robot move on a certain track, usually marked by a black line. Multiple types of sensors can be used to steer the robot, such as the light sensor and the IR sensor. A camera can also be used in complicated devices to achieve the same goal with better accuracy.

In this project, the user will learn how to use the IF statement to steer the robot in two different directions based on two IR sensor readings.

Look at Page 40 to learn more about IF statement.
To make a line following robot in a better way, at least three sensors should be used. This is to control the motors more accurately. This way, the robot will move straight when the line is only detected by the middle sensor. Similarly, the robot will move right when the line is under the right sensor and will move left when the line is under the left sensor.

In this project, the user will learn how to program a line follower robot with three IR Sensors, depending on the following table:

<table>
<thead>
<tr>
<th>IR1</th>
<th>IR 2</th>
<th>IR 3</th>
<th>RM</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>Imposable case</td>
<td></td>
</tr>
</tbody>
</table>
Controlling Motors Using Bluetooth

Bluetooth: is a communication method which uses electromagnetic waves to transfer data. Bluetooth uses a frequency of 2.4 MHz, and works only over short distances, no more than 100 meters. This allows Bluetooth to be energy efficient. Bluetooth is widely used in earphones and smartphones. The Ebot contains a Bluetooth device which works over distances as far as 10 meters. This allows the user to control the Ebot’s inputs and outputs remotely. Furthermore, the user can program any project through the Ebot Blockly App available on (OSxAndriod).

In this project, the user will learn how to control motors remotely using the Bluetooth remote available in the Ebot Blockly App. Directions are based on two IR sensor readings.

Implementation Steps

1. Set up the Ebot to use Bluetooth
2. Connect the project and program it as shown.
3. Connect the Ebot using USB and turn it on.
4. Click on the Download button.
5. After downloading the app and starting it, click on to find your Ebot.
6. Click on and start controlling your project.

General Notes:

If gears were connected to the motor, the tire movements would go in the opposite directions. Therefore, it is necessary to flip the DC Motor wires for it to move correctly.

Look at Page 44 learn how to set up the Ebot to use Bluetooth.
Learn to Use the LCD

Liquid-Crystal Display (LCD): is a single colored screen used to show text messages (generally in English). Different LCD screens have a different number of lines and characters that can be utilized. The LCD screen available with the Ebot has the ability to display 16 characters per line, and contains a total of two lines, allowing for messages up to 32 characters in length. There are two fields contained within the settings of the LCD screen; they are:

- **Row**: As mentioned earlier, the Ebot’s LCD contains two rows. The user can choose between the upper row (Row 0) and the bottom row (Row 1).
- **Column**: Used to specify which column in the chosen row to begin typing the message.

In this project, the user will learn how to use the LCD to show a text message when a Button is clicked.

### Implementation Steps

1. Connect the project as shown.
2. Click on the (g) to show the block settings, then choose the options as described below:
   - Type the word you want to show in the field next to the word Text
   - To write in the second row, change the value of the field in the next Row to 1
   - Change the Delay in the second field to 1000.
   - Unchecked Clear LCD
3. Connect the unit, turn it on, and click the Download button.

### General Notes:

The LCD has several settings that can be used:

- **Clear LCD**: This allows the screen to be cleared before the new text is written.
- **Delay**: Delay before showing the message, which can be between 50-5000 milliseconds, based on what the user specifies.
Controlling RGB

RGB is a light that can generate any visible light using only one device (details in Project 3). This unit combines different intensities of the three primary colors (Red, Green, Blue) to generate the color specified by the user. Similarly, sending black to the RGB will turn off all the lights within it.

In this project, the user will learn how to control the RGB using two buttons to generate to different colors on the RGB.

Implementation Steps

1. Connect the project as shown.
2. Drag the Button from the Input menu and the RGB from the output menu and place them as shown.
3. Click the () blocks and change their settings to match the code as shown.
4. Connect the Ebot using USB and turn it on.
5. Click the Download button.

General Notes:

Notice the role of the delay in the way the light appears when its value is changed!

Can you make a traffic light that works using a single button?
Using the Ultrasonic Sensor

Ultrasonic Sensor: Ultrasonic waves that move through matter (Water - Air - Solid Objects), and it caused by movements of bodies, appearing as vibrations. An Ultrasonic Sensor is made up of two parts: a transmitter and a receiver. The transmitter sends ultrasonic signals, and the receiver receives the ultrasonic signal bounced back from solid objects.

In this project, the user will learn to control the Ultrasonic Sensor and send a voice and visual alarm when an object is detected at a specified distance.

Implementation Steps

1. Connect the project as shown.
2. Drag the Ultrasonic Sensor from the Input menu and the Buzzer and LED from the Output menu and place them as shown.
3. Click the ( ) and make sure the settings match the ones show.
4. Connect the Ebot using USB and turn it on.
5. Click the Download button.

Do you know that the contains a delay in its block and the can be replaced by modifying the delay setting of the Buzzer.

Try and notice the difference!
Project 14

Using the Repeat

Repeat: is a programming feature provided by the Ebot Blockly software to simplify the process of repeating orders multiple times for users.

The Repeat block can be used in two ways:

- **Limited Repeat**: Works by specifying how many times the orders will be repeated. This is done by using 0 from the Variables menu and entering the number of times.

- **Forever Repeat**: This will repeat the code inside the block infinitely or until the Break block order is encountered.

In this project, the user will learn how to use the Repeat block to create a light and sound alarm.

Implementation Steps

1. Connect the project as shown.
2. Drag the blocks from their menus to the screen as shown.
3. From the Flow menu, drag the repeat as shown and unchecked the Forever setting.
4. From the Variables menu, drag the 0 to the space next to the Repeat and enter the value 3.
5. Connect the Ebot using USB and turn it on.
6. Click the Download button.

There are different ways to repeat code in programming. The most important are the For loop and the While loop. The While loop is frequently used when there’s a condition that needs to be satisfied in order to repeat. Example:

```c
while (digitalRead(A0) == HIGH) { Buzzer(); }
```

The For loop, however, depends mainly on a certain number of repetitions tied to a variable.

General Notes:
Using the Potentiometer

Potentiometer is a component that allows the user to change its resistance by turning a knob. This kind of device is used in devices that depend on changing the Work it’s exerting and are classified as Analog Devices. Resistance has an inverse relationship with Work. If the resistance value decreases, the Work value increases. Similarly, if the resistance value increases, the Work value decreases and can go down to 0.

In this project, the user will learn how to use the Variable Resistor and the Serial Monitor to send text to the computer expressing the value of the current resistance.

Implementation Steps

1. Connect the project as shown.
2. Drag the blocks from their menus to the screen as shown.
3. Click the (1) in the Serial TX and then type the text to be sent in the empty box after Text word.
4. Connect the Ebot using USB and turn it on.
5. Use the Input Reading feature to change the IR reading values, then click .

General Notes:

The Serial Monitor screen is the connection between the Ebot and the computer, where the user can use it to send and receive control commands to and from and to the computer. We can open the Serial Monitor from the left menu and check the messages that are sent to the computer by the Ebot when the value of the Variable Resistor changes.
**Potentiometer** is a component that can be used in many different creative ways. This is because the user can directly control the amount of resistance within the component’s circuit. It allows the user the capability of executing different commands based on the value of the resistance. Some of these commands can be:

- Controlling the intensity of LEDs
- Controlling the speed on a DC motor
- Controlling the angle of the servo

In this project, the user will learn to split the readings of the variable resistor into 4 sections to control LEDs sequentially. This way, every section will execute different commands based on the reading of the Potentiometer.

**Implementation Steps**

1. Connect the project as shown.
2. Drag the blocks from their menus to the screen as shown.
3. Click the LED and then drag the NextOutput to under Output to add another pin for the LED. Repeat it twice.
4. Click the () then select the pin numbr, and its state for each LED.
5. Connect the Ebot using USB, and turn it on, and click .

**General Notes:**

To remove one of the LED from the group, click on and drag the NextOutput block outside the square or put it back in its place on the left.
Controlling Outputs with Custom Code

Custom Code: is a feature that allows the user to add commands and comments without leaving the Graphical User Interface (GUI). This feature hence allows for developing custom properties for the sensors and outputs. There are two types of Custom Code that can be added through the GUI.

- **Comment**: Comments are used to explain what specific sections of the code does, or what needs to be added to the code later, without interrupting the program execution.
- **Plain Text**: Used to add code to be executed within the program.

In this project, the user will learn how to use Custom Code to control the outputs by changing the value of the Variable Resistor (Pot).

**Implementation Steps**

1. Connect the project as shown.
2. Drag the blocks from their menus to the screen as shown.
3. Add a variable named POT by clicking setting its value to be 0 and then click Create.
4. Click the ( ) in Custom Code and choose Plain Text, then enter the code as shown to the left.
5. Connect the Ebot using USB and turn it on.
6. Click the Download button.

**General Notes:**

The C++ programming language is a case-sensitive language. This means it considers lowercase and uppercase letters as separate letters (eg. analog is not the same as Analog). When writing commands with the wrong letters, the program will not execute those commands because it reads them differently.

Ensure that the case of the letters match in all your commands to avoid such errors.
**Using the Temperature Sensor**

**Temperature Sensor**: is a device of type (LM35) used to measure the temperature of the environment surrounding it. This is done by changing the voltage of the sensor based on the temperature. It is worth noting that the temperature does not need to be setup for the environment surrounding it and the range of temperatures is between -55 to 150 degrees Celsius, with a 0.05% margin of error.

In this project the user will learn how to program the Temperature Sensor to turn on the motor when the temperature rises.

**Implementation Steps**

1. Connect the project as shown.
2. Pull the temperature sensor from the Input menu and the DC motor from the output menu and place them on the Graphical Page as shown.
3. Click the (i) in the sensor and choose the properties as shown.
4. Connect the Ebot using USB and turn it on.
5. Click the Download button.

**General Notes:**

It is important not to allow the temperature sensor’s surrounding temperature to exceed 150 Celsius, as that would lead to breaking the device.

The sensor should never be fed a voltage above 9 volts as that would lead to incorrect readings.
Create a Digital Thermometer

**Variables:** is one of the most important programming concepts, as it reserves parts of memory under a name chosen by the user. These variables contain information and values that could change during the execution of the program. Variables can be described as jars, where a value is placed in them. This value is kept and can be read or changed when needed. Variables can contain letters or numbers.

In this project, the user will program the LCD to show the temperature in the surrounding environment, which is done using a variable, and using a formula to transform the value of the sensor from byte to a value that describes the temperature in Celsius.

**Implementation Steps**

1. Connect the project as shown.
2. Drag the blocks from the left menu to the Graphical Page.
3. Add a variable with the name of (temp) by clicking on in the left menu, setting its value to be 0, and clicking Create.
4. Click the (f) in Custom Code and choose (Plain Text) then type the programming as shown.
5. Connect the Ebot using USB and turn it on.
6. Click the Download button.

**General Notes:**

Some of the frequently used variable types:

- **char:** letter
- **byte:** a byte of memory
- **int:** a number composed of 4 bytes
- **Long:** a large number consisting of 8 bytes
- **Float:** a number that can handle decimals
- **Double:** a large number that can handle decimals.
Creating a Color Reader

**Color Sensor**: is a component used to measure the intensity of colors. Color Sensors depend on the frequency of light reflected by different bodies when exposed to a white light. Color sensors contain three different sensors, one to sense each of red, green, and blue. It is worth noting that combinations of these three colors can generate any color.

In order to get good and accurate readings, the user should carefully choose the distance between the sensor and the object. They should be close to each other, but not too close. If the object is far, the object is touching the sensor, the white light will no longer reflect off of the object into the sensor and the color would not be identified correctly.

In this project, the user will learn how to use the Color Sensor to read the color and show it on the screen.

**Implementation Steps**
1. Connect the project as shown.
2. Drag the blocks from the left menu to the Graphical Page.
3. Click the ($) in the sensor block and set the properties as shown.
4. Connect the Ebot using USB and turn it on.
5. Use the Input Reading $ feature, and change the range of values to be detected accordingly. Click the Download button.

**General Notes:**
Sensors, in general, are affected by the environment surrounding them (light intensity, color level, paper type, etc.). Therefore, it is important to use the Input Reading $ feature to change the values that the sensor will detect based on the surrounding environment.
Multimedia: is an important feature in educational projects and interactive games and shows. The user is able to further improve programs by allowing the program to play different types of Multimedia such as audio, images, and videos. There is also a feature to control the Ebot using the Keyboard and Mouse (available on PC).

In this project, the user will learn how to develop a program to show different multimedia based on the readings of the color sensor.

Implementation Steps

1. Connect the project as shown.
2. Drag the color sensor from the output menu on the left, and the type of media you wish to describe from the Multimedia menu to the Graphical Page.
3. Click the (f) in the color sensor and choose the properties as shown. In the Multimedia block, click on Browse and select the file to be played.
4. Add a Stop command to the other Multimedia when switching to another one.
5. Connect the Ebot, then Click .

General Notes:

All Multimedia blocks all contain Play, Stop, and Browse commands.

The Ebot Blockly program only supports audio encoded in MP3. The program also depends on the video encodings supported in the PC the program is running on. Therefore, make sure that the PC can run the video before adding it to the program.
Using the BreadBoard

**BreadBoard**: is a plastic board with many holes used to conduct tests and experiments with electric circuits without the need to solder components to it. It depends on placing and removing components directly, which saves a lot of time and effort in the early stages of developing a prototype for a project.

Head holes are connected with each other within a BreadBoard, so the user should connect components horizontally to connect the ends of a component.

In this project, the user will learn how to use the BreadBoard to build an electric circuit and connect it to the Ebot.

---

**Implementation Steps**

1. Collect the following components:
   - LED
   - 10K Ω resistor
   - Male-to-Female Cable.
   - Tiny BreadBoard
2. Drag the LED from the output menu and place it as shown.
3. Connect the Ebot through USB and turn it on.
4. Click on the Download button.

---

**General Notes**: The LED contains two sides, an anode (positive) and a cathode (negative). The anode has a long wire and the cathode has a short one. Make sure they are connected correctly so that the white wire is connected to the positive and the black wire is connected to the negative.

Try programming a blinking LED!
Using the IF Statement

**IF Statement**: is one of the most important concepts in all programming languages. It is used to execute commands IF a condition is true. A program can have multiple IF statements.

The Ebot system allows for four additions to the IF statement. These show when the □ is clicked:

- **AND**: All conditions must be met.
- **OR**: One or more of the conditions must be met.
- **ELSE IF**: Checked if the original IF is false.
- **ELSE**: Executed if none of the IF / ELSE IF statements’ conditions are met.

In this project, the user will learn how to use the IF statement to turn on an LED only if the Button is pressed.

**Implementation Steps**

1. Collect the following components:
   - LED
   - Button
   - Two resistors (220 Ω and 10K Ω)
   - Male-to-Female cables
   - Tiny BreadBoard

2. Drag the LED from the Input, then click the □ next to it and pull the ELSE to under the IF.

3. Connect the Ebot through USB and turn it on.

4. Click on the Download button.

**General Notes:**
Program Using the C++ Programming Language

The Code Page is the area that the user can use to program his project using a programming language such as C++. It also allows the user to copy any code written in C++ and paste it in the program to execute on the Ebot.

Warning: If the user wishes to program the project using the Code Page, there should not be anymore changes or additions in the Block Page, because that would cause the code written to be removed.

In this project, the user will learn to use the Code Page to program the previous project.

Implementation Steps:

1. Reconnect the previous project.
2. Search 'Arduino Button' on Google.
3. Open the page and copy the Code section.
4. Open Ebot Blockly and do the following:
   i. Select the Code Page
   ii. Remove everything after the '#include Ebot.' and paste the code that you copied.
   iii. Change the value of the pins to (inPin = A0) and (ledPin = 0))
5. Connect the Ebot through USB and turn it on.
6. Click on the Download button.

General Notes:

Look at Page 48 for more information about the C++ Programming Language.
Chapter Three: Extra Features
Implementation Steps:

1. Open Ebot Blockly on your smartphone/tablet.
2. Turn on Bluetooth and Location in your phone.
3. Turn on the Ebot either through USB or an external battery.
4. Click on the Bluetooth button in the Ebot app and choose your Ebot.
5. The program will automatically recognize the Ebot and you can then use it by clicking on at the top of the page.

Connecting the Ebot with Your Device

In the Ebot Blockly software, the latest technology is used to connect the Ebot to the software wirelessly. It gives the user the ability to create Ebot programs from anywhere. To do so, the following steps must be performed:

Implementation Steps:

1. Open Ebot Blockly on your smartphone/tablet.
2. Turn on Bluetooth and Location in your phone.
3. Turn on the Ebot either through USB or an external battery.
4. Click on the Bluetooth button in the Ebot app and choose your Ebot.
5. The program will automatically recognize the Ebot and you can then use it by clicking on at the top of the page.

Look at Page 13 to learn how to Download Ebot Blockly Software.
Implementation Steps:

1. Connect the sensor to one of the Analog Input pins, and make sure that the Ebot is connected to the computer and turned on.
2. In the Ebot Blockly software, click on Input Reading in the menu on the left.
3. Select the type of sensor and the pin that is used to connect to it, and then click Debug. Make a note of the values.

Custom Input

This block is considered one of the most important inputs in the Ebot Blockly. It allows the user to use many different sensors without the need to program them, as it depends on the Analog-to-Digital Converter, which transforms the signal from an unreadable Analog signal to a readable digital one.

The Custom Input settings also allow for an important feature which is the detection of the Rising and Falling edges of a signal. This feature is used when outputs need to be controlled when the signal changes rather than the analog value of the sensor. Custom Outputs can also be programmed using the Ebot Blockly. They can be given a value of 5 volts for high or 0 volts for low.

With the advancement of digital technology, it has become necessary to find new states to cope with large amounts of information. This gave birth to the new states of Negative (Falling) Edge and the Positive (Rising) Edge. The states Negative Edge and Positive Edge are considered instant states that only happen when the signal changes, while 0 or 1 are constant states.
Live Control

Live Control is an extra feature that saves a lot of time and effort by allowing the user to connect directly to the outputs and control them, without the need to download the code to the Ebot. The capabilities of this feature are as follows:

1. Connect the Ebot with the outputs that you wish to control connected to it.
2. Click on 🔄 from the left menu.
3. Select the output pin and then select the type of output from the drop-down.
4. Repeat (3) for all the outputs then click Start.
5. You can change the output settings and watch them happen immediately.
6. You can control the motors from the top right of the Live Control screen.
7. The user can also try several states for one more of the outputs. This is done using the Add State button to add as many states as needed. Once completed, click the Send All States button to see the states executed consecutively.
8. Move the final settings to the main program. *

* Live Control is only used as a measurement tool and not as final programming.

Implementation Steps:
Data Lab

Data Lab is an extra feature added to the Ebot Blockly software used to read and analyze the data by recording the readings for inputs for a specified time. This data can be analyzed and studied and exported to Microsoft Excel.

Implementation Steps:

1. Connect the Ebot with the outputs that you wish to control connected to it.
2. Click on from the left menu.
3. Specify the following details:
4. Select Pin: Choose the pin that the input is connected to in the Ebot.
5. Number of Readings: Specify the total number of readings to be recorded.
6. Read Every: Specify the time between each reading.
7. Beep on Values bigger than: Generate a sound when the reading exceeds a certain value
8. Data: Through the Data feature, a mathematical equation can be set up to be applied to the readings
9. Connect: Begin reading the data
10. After reading enough data, click (Save to Excel File) to export the data to an Excel file.
11. Open the exported file using Microsoft Excel and then go to the Insert tab. Click (Recommended Charts) to get the best chart type for the data specified. For more information about charts, visit (www.microsoft.com) and search for (Create a Chart).
C++ Programming Language

C++ is a multipurpose programming language, considered a completed version of the older programming language C. Considered by many to be the best programming language to design complex interfaces, C++ has fast execution times and is considered a high-level programming language. At the same time, it is similar to low-level languages such as Assembly code can be merged with C++ code. This language is used in typing the commands for Ebot Blockly, and the most important commands are listed below:

- **int**: An abbreviation for the word integer. int is a variable that can take on any whole number. It is worth noting that there are other types of variables.
- **void setup()**: A function that executes that commands inside it on the execution of a program, and it is used to specify the settings of inputs and outputs, and declare variables.
- **pinMode()**: Specify whether the pin is going to function as an input or as an output.
- **void loop()**: Contains the function part of programming and is repeated until the device is turned off.
- **delay()**: This command is used to delay the execution of the program for a specified time calculated in milliseconds.

What is Arduino?

Arduino is an electronic open-source platform that is used in prototyping electronics using easy to understand hardware and software. Open-source refers to the platform schematics and software code available freely worldwide, making it easy for anyone to modify and/or create their own open-source application.

The inventors of the Arduino created an Integrated Development Environment (IDE) to help people easily program and develop applications. Arduino’s IDE uses C++ and is marked by its easy-to-understand platform and its huge support library, which contains sample projects and lessons designed to introduce beginners to Arduino. This library can be found at: https://www.arduino.cc

The C++ language is case-sensitive. Commands, therefore, must be written correctly in order for the program to work. (analogread is not the same as analogRead ).
**Ohm’s Law**

**Electric Current** is a flow of free electrons from the positive end (+) of a circuit to the negative end (-).

This flow contains a voltage that can execute work. As voltage increases, the amount of work that can be executed by the electric current increases as well. To measure voltage, Ohm’s law (V) must be used. **Ohm’s Law** describes the relationship between the **voltage (V)**, the **resistance (R)**, and the electric **current (I)**.

This law is summarized in the following equation: \( V = I \times R \)

We can use this law to calculate the maximum amount of electric current that is allowed to pass through any electrical component without burning it. We will use the LED as an example. The information provided to us is:

1. An LED needs a voltage of 1.6 volts to turn on.
2. The signal pin of the Ebot provides 5 volts.
3. The voltage at the GND pin is always 0v, so we can conclude that the voltage is coming from the signal pin.
4. **Signal Voltage = LED Voltage + Resistor Voltage**
5. Using Ohm’s Law we can conclude \( I = (V/R) = (3.4/220) = 0.015A = 15mA \)

**Table:**

<table>
<thead>
<tr>
<th>Voltage (V)</th>
<th>Resistor (Ω)</th>
<th>Current (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5.6KΩ</td>
<td>15mA</td>
</tr>
</tbody>
</table>

**Ohms Law:**

\[ V = I \times R \]
Chapter Four: Obstacle Avoiding Car
Obstacle Avoiding Car Project

The Obstacle Avoiding Car is a technique that depends on the robot sensing physical bodies in front of it through the use of sensors (Ultrasonic, IR, etc). Based on the values of these sensors, the robot can reverse then circle around the object that it sensed. This technique is currently used in the robot vacuum, which can clean rooms without the need of removing furniture or the obstacles in front of it.

In this project, the user will learn how to make an obstacle avoiding car in its simplest forms by using previous programming experience in addition to the plastic blocks to build a fully operational prototype.

The program starts by waiting for the button to be clicked. When clicked, the program will enter a loop and begin reading the distance to the objects in front of it using the Ultrasonic Sensor. The LED in this stage is a green light. When an object reaches close proximity of the car, the LED will turn red and an alarm will sound. The robot will then change its direction. This process is repeated for as long as the robot is operational and until the user clicks on the button a second time.
1. Start by moving the pole’s shaft from the middle to the far right or far left to be able to position the two DC motors side by side. Push up the back part of the motor to open the clear part.

2. Remove the shaft with the gears out of the motor. Push the gears in one motor to the far right and the others to the far left. Put the shaft and gears back in the motor and close the clear plastic part.

3. Make sure that the direction of the shaft is reversed in the two motors, one to the right and the other to the left.

4. Place the two motors on the 9x15 plastic board and make sure that the wires are close to the shorter edge of the plastic board. Place the L-Adapter in the middle between the two motors as shown. Place a 3x5 plastic board on a Small Frame piece and then place it on the 9x15 board as shown.

5. Use two pieces of the 3x15 board and place them on the sides of the project as shown. Press on the piece until you hear a clicking sound that ensures that the piece is attached correctly.
6. Use two pieces from the Full Bush and Small Gear and attach them to the motor shaft. The Full push should be attached first followed by the small gear.

7. To make the wheel, attach the following pieces as shown: Wheel-Large Gear-Large Shaft-Full Bush.

8. Attach a 3x5 plastic to one side and attach the wheel in a way that it fits with the small gear, then add a full bush to hold it tightly.

9. The final shape of the bottom part of the project should be as shown.
10. Flip the project and use 3 pieces of the 3x5 plastic and attach them as shown.

11. Use two pieces of the 1x5 and attach them to the bottom of the Ebot, then place the Ebot on the 9x15 board.

12. Attach the button to the back piece of the project as shown, then connect it to pin A0 on the Ebot.

13. Attach the Ultrasonic sensor to the front part of the project then connect it with pin A1, attach the Buzzer and RGB above the Ultrasonic Sensor as shown, and connect the Buzzer to Pin 1 and the RGB to Pin 2, then start programming the project.
Project programming

Implementation Steps

1. Connect the project as shown.
2. Drag the button and ultrasonic sensor from the input menu. Drag the LED, buzzer, and DC motor from the output menu to the Graphical Page.
3. Click the (i) in the components in the Graphical Page, then choose the properties as shown in the programming. While programming, try to imagine the way the program will operate.
4. Connect the Ebot using USB and turn it on.
5. Click the Download button

Use previous similar projects for help: 10 - 13

RGB1
RGB4
RGB2
RGB3
3R3
RGB
Get Certified by Testing Your Knowledge..

After completing all projects within this guide, you are now qualified to get a Certificate of Experience in the Ebot system. To get your certificate, you have to pass the test on the website and print your certificate online.

Serial Number: 625377

Enjoy and Improve Your Skills..

Scientific innovation stories is a series of stories about young adventurers facing problems in life and inventing solutions using Ebot. Each book contains four different scientific technology projects for children that develop programming, electronics connections, and mechanical skills.

40 Scientific project

10 Books

onc.makersacademy.net
Meet the innovator kids..

Educational stories about technological innovations

(D) (Sara)

(E) (Khalid)

(X)