

Design And Implementation Of A Robotic Arm Based On Haptic Technology

¹ A. Rama Krishna ² G. Sowmya Bala ³ A.S.C.S. Sastry
⁴ B. Bhanu Prakash Sarma ⁵ Gokul Sai Alla

¹ Assistant Professor, Dept. of ECM. K.L.University, Vaddeswaram, A.P, India,

² PG scholar, Dept. of ECM. K.L.University, Vaddeswaram, A.P, India,

³ Professor, Dept. of ECE. Srinivasa institute of Engineering & Technology,

^{4,5} B-Tech(IV/IV), Dept. of ECM. K.L.University, Vaddeswaram, A.P, India,

ABSTRACT

The present manuscript deals with designing a haptic robotic arm, which can be used to pick and place the objects. In this paper, a robotic arm with four degrees of freedom is designed and is able to pick the objects with a specific weight and place them in a desired location. To facilitate the lifting of the objects, Servomotors with a torque of 11 kg are used. The programming is done on ATMEGA-328 Microcontroller using Arduino programming. The Microcontroller along with input pins is soldered on a PCB board. The input is given using a remote, which is an arm, made of Polycarbonate fitted with Potentiometers with a certain angle of rotation. The Potentiometers detect the angle of the rotation and the signals are sent to the Microcontroller accordingly. In the world of robotics, this Robotic arm has been turned out to be trendy. This kind of the arms have many applications in the field of industrial robotics where the automation is required.

Keywords: *Haptic, Servo Motors, ATMEGA-328 Microcontroller, Arduino, Potentiometer.*

1. INTRODUCTION

A robot is an automatic or virtually intelligent agent that can carry out tasks robotically or with some supervision, typically with the aid of a remote control. In practice, a robot is usually an electro-mechanical machine that is guided by means of computer and electronic programming. Robots can be autonomous, semi-autonomous or remotely controlled [1]. Robots are used within an increasingly wide variety of tasks such as for household appliances like vacuuming floors, mowing lawns, cleaning drains, building cars, in warfare, and in tasks that are too expensive or too dangerous to be performed through humans such as exploring outer space or at the bottom of the sea.

In the implementation process, the software consists of the commands that control a robot's actions and provide information regarding required tasks. When a program is written by means of software, the robot is able to implement commands and achieve the particular errands [4]. Programming robots can be a intricate and difficult process, and while it has become easier over the years, the lack of cross-platform industry principles has

affected the development of software tools for robots compared to other automated control systems such as programmable logic controllers (PLCs) [4].

Haptic technology, or haptics, is a tactile feedback technology, which takes advantage of the sense of touch by applying forces, vibrations, or motions to the user. This mechanical stimulation can be used to assist in the creation of virtual objects in a computer simulation, to control such virtual objects, and to enhance the remote control of machines and devices (telerobotics). It has been described as "doing for the sense of touch what computer graphics does for vision" [2]. Haptic devices may incorporate tactile sensors that measure forces exerted by the user on the interface. This technology has made it possible to investigate how the human sense of touch works by allowing the design of cautiously controlled haptic virtual objects. These objects are used to steadily probe the human haptic capabilities, which would otherwise be difficult to achieve. These research tools contribute to the understanding of how touch and its underlying brain functions work [1]. Robotics involves elements of mechanical and electrical engineering, as well as control theory, computing and now artificial intelligence (Selig, 1992). According to the Robot Institute of America, "A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks" [1]. The robots have to interact with their environment, which is an important objective in the development of robots. This interaction is commonly accomplished by means of some sort of arm and gripping device or end effector's [3]. In the robotic arm the arm has a few joints, similar to a human arm, in addition to shoulder, elbow, and wrist, coupled with the finger joints; there are many joints [2].

The design process of is clearly explained in the next section with detailed information regarding the components used then is followed by the implementation and the results and finally ends with conclusion.

2. DESIGN OF THE ROBOTIC ARM

The Robotic Arm is designed using the Microcontroller i.e. ATMEGA - 328 Micro-controller using

Arduino programming. This works on the principle of interfacing potentiometers and servos.

This is done using Arduino Board. The remote is fitted with potentiometers and the servos are attached to the body of the robotic arm. The potentiometer converts the mechanical motion into electrical motion. Hence, on the motion of the remote the potentiometers produce the electrical pulses, which are en route for the arduino board. The board process the signals received from the potentiometers furthermore convert them into requisite digital pulses that are then send to the servomotors. This servo will respond as per the pulses and the moment of the arm occurs. In short, the micro controller interfaces all these components specified above.

A short list of it as follows:

1. Servo motors
2. Potentiometers
3. Atmega-328
4. Arduino duemilanove

2.1. Servo motors

A servomotor is a motor, which forms part of a servomechanism. The servomotor is paired with some type of encoder to provide position/speed feedback [7]. A stepper motor is one type of servomotor. A stepper motor is actually built to move angular positions based upon each possible step around the entire rotation, and may include micro-steps with a resolution such as 256 micro-steps per step of the stepper motor.

A servomechanism may or may not use a servomotor. For example, a household furnace controlled by a thermostat is a servomechanism, because of the feedback and resulting error signal, yet there is no motor being controlled directly by the servomechanism.



Figure 1 A ServoMotor

Figure 1 shows the image of a servomotor. Figure 2 shows the interlinks and the working of the servomotor interfacing with the other components. It is clearly shown that an encoder or resolver positive feedback signal is sent to the digital controller from the servomotor. In addition, a velocity feedback signal is sent

to the corresponding amplifier to provide an efficient functioning of the servomotor.

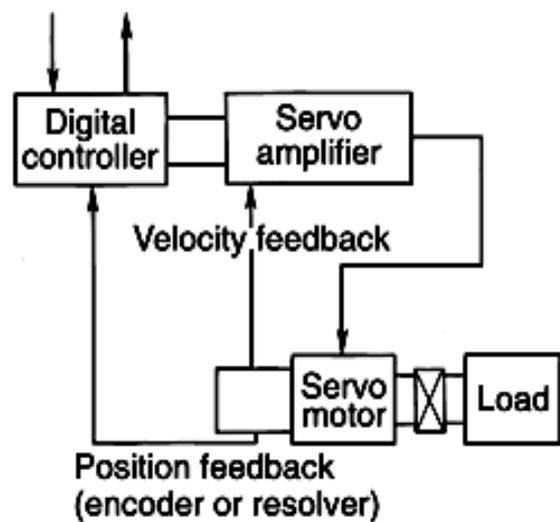


Figure 2 Servomotor interfacing with Amplifier and Digital Controller

Figure 3 shows the circuit diagram of the servomotor built out of five resistors and a capacitor, aided by a transistor. The power and control are given as input to this circuit.

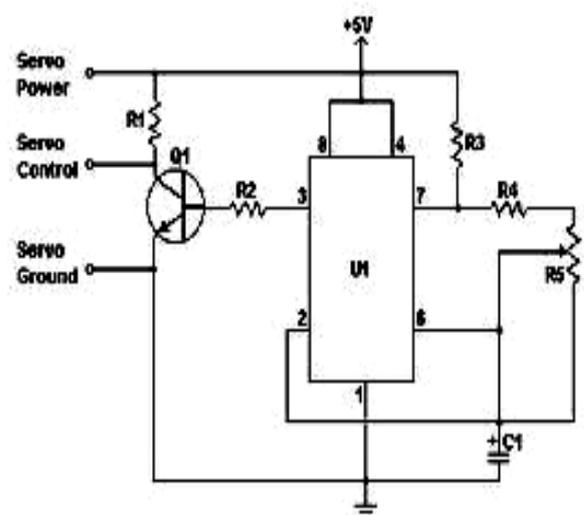


Figure 3 Circuit diagram for servomotor

2.2. Potentiometer

A potentiometer informally, a pot, within electronics technology is a component, a three-terminal resistor with a sliding contact that forms an adjustable voltage divider [7]. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.



Figure 4 A Potentiometer

A potentiometer is essentially a voltage divider used for measuring electric potential (voltage) [7]; the component is an implementation of the same principle, whence its name. Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick [7].

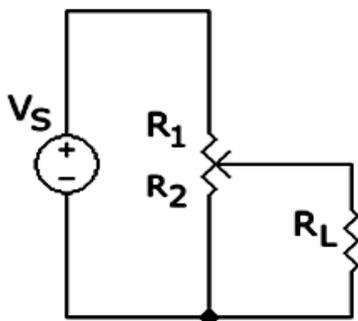


Figure 5 Circuit diagram of a potentiometer

2.3. Arduino duemilanove

The most important parts on the Arduino board are highlighted in red in the figure 6 below and are as follows:

- USB connector
- Power connector
- Automatic power switch
- Digital pins
- Analog pins
- Power pins
- Reset switch

Arduino is an accepted open-source single-board microcontroller, successor of the open-source wiring platform designed to formulate the process of using electronics in multidisciplinary function more easily to get [6].

A simple open hardware design for the Arduino board with an Atmel AVR processor and on-board input / output support [5] is used in the hardware part of the design process. On the other hand, programming language compiler, which is a standard one and the boot loader are there with in the software that runs on the board.

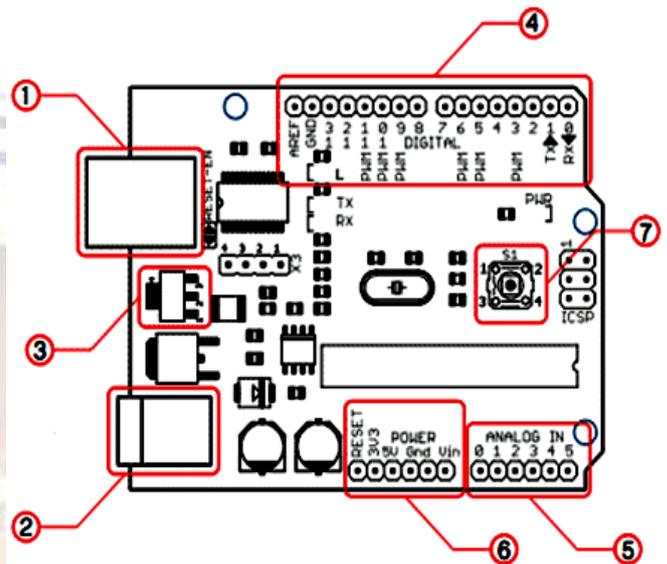


Figure 6 Arduino duemilanove circuit top view

With the help of the wiring-based language (syntax and libraries) similar to C++, Arduino hardware is programmed using some simplifications and modifications, and a Processing based integrated development environment.

3. IMPLEMENTATION PROCESS

3.1. Working Principle

Figure 7 shows the complete circuit and the working principle of it is as follows.

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

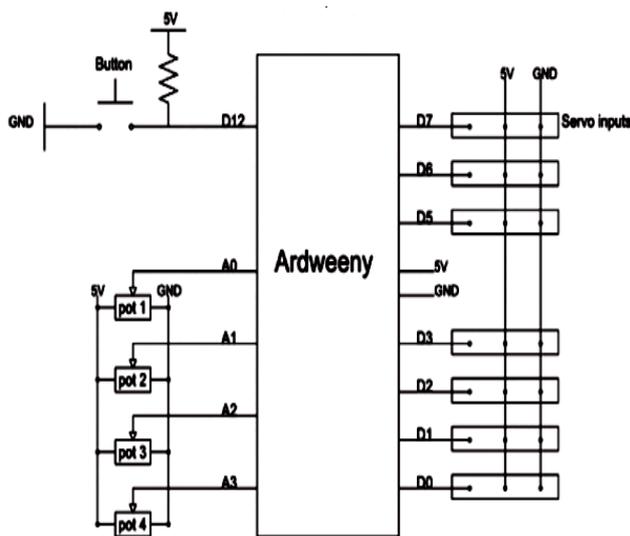


Figure 7 Complete Circuit Block Diagram

Each of the 14 digital pins on the Duemilanove can be used as an input or output, using the command lines 'pinMode()', 'digitalWrite()', and 'digitalRead()' as functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions.

There is a built-in LED connected to digital pin 13. When the pin is "HIGH" value, the LED is on, when the pin is "LOW", it is off. The Duemilanove has six analog inputs, each of which provides 10 bits of resolution (i.e. 1024 different values).

3.2. ATmega328 Microcontroller

3.2.1. General Description

The ATmega88 through ATmega328 microcontrollers are the upgrades from the ATmega8. They are pin compatible, but not functionally compatible [5]. The ATmega328 has 32kB of flash, where the ATmega8 has 8kB. Other differences are in the timers, additional SRAM and EEPROM, the addition of pin change interrupts, and a divide by 8 prescaler for the system clock. The schematic below shows the Atmel AVR ATmega328 circuit as it was built on the test board.

The power supply is common and is shared between all of the microcontrollers on the board. The ATmega328 is in a minimal circuit [5]. It is using its internal 8 MHz RC oscillator (divided by 8). The boot loader is programmed using the ISP programming connector, and the Arduino sketches are uploaded via the 6-pin header. The crystal shown on the schematic is required when the ATmega328 is going to be used as an Arduino, although it may be desired in any real world application [5].

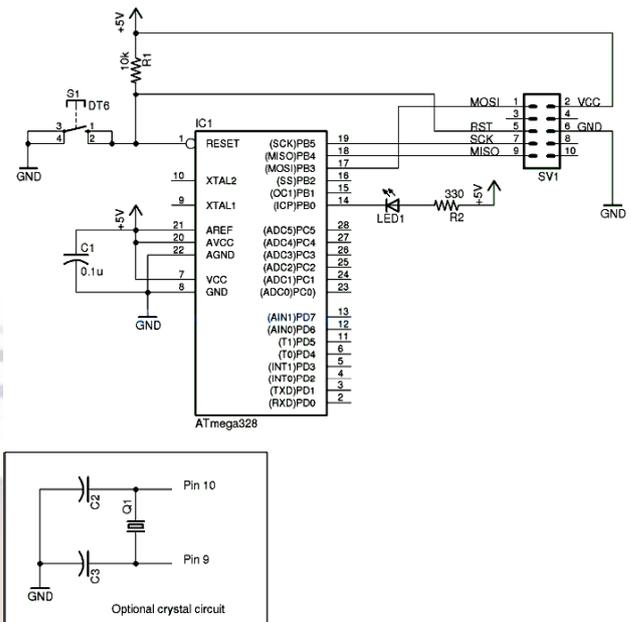


Figure 8 Schematic View of Atmel AVR

The internal parts that are used are as follows.

1. IC1 Atmel ATmega328 MCU
2. C1 0.1 uF ceramic capacitor
3. C2, C3 22 pF ceramic capacitor
4. Q1 16 MHz crystal
5. R1 10k 5% 1/4W resistor
6. R2 330 5% 1/4W resistor
7. LED1 5mm red LED
8. S1 Momentary pushbutton switch
9. SV1 10 pin shielded header or dual row header
10. X1 6 pin Molex KK100 or single row header
11. Perf board
12. 28-pin socket
13. Power supply

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is for the boot loader); the ATmega328 has 32 KB, (also with 2 KB used for the boot loader). The ATmega168 has 1 KB of 'SRAM' and 512 bytes of 'EEPROM' (which can be read and written with the EEPROM library) the ATmega328 has 2 KB of 'SRAM' and 1 KB of 'EEPROM'.

By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the "AREF" pin and the 'analogReference()' function. External Interrupts (2 and 3) are the pins that can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

The 'attachInterrupt()' function is used for this purpose. PWM (3, 5, 6, 9, 10, and 11) are provide with 8-bit PWM output with the 'analogWrite()' function. SPI (10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)) are the pins that support SPI communication using the SPI library.

Brief summary of this is as follows

Microcontroller	:	ATmega168
Operating Voltage	:	5V
Input Voltage (recommended)	:	7-12V
Input Voltage : (limits)	:	6-20V
Digital I/O Pins	:	14 (of which 6 provide PWM output)
Analog Input Pins	:	6
DC Current per I/O : Pin	:	40 mA
DC Current for 3.3V: Pin	:	50 mA
Flash Memory	:	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	:	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	:	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	:	16 MHz

3.3. Communication

The Arduino Duemilanove has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide 'UART TTL' (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An 'FTDI FT232RL' on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor, which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Duemilanove's digital pins. The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a wire library to simplify use of the I2C bus. For SPI communication, the SPI library is used [6].

3.3.1. LED (Light Emitting Diode)

A light-emitting diode (LED) as shown in the figure 8, is a semiconductor light source that is used as indicator lamps in many devices and are increasingly used for other lighting [7]. When a light-emitting diode is forward biased (switched on), electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

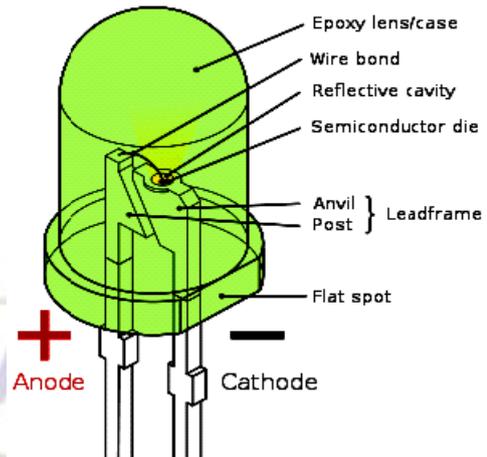


Figure 9 LED

This effect is called electroluminescence and the energy gap of the semiconductor determines the color of the light (corresponding to the energy of the photon) [7]. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, and faster switching [7]. Bring the reset line 'LOW' to reset the microcontroller. Typically, this is used to add a reset button to shields, which block the one, on the board.

3.4. Applications

This Haptic Robot arm is used for various applications, of them few are as follows:

- To lift heavier objects
- To lift nuclear wastes without harming the humans
- Used as external limbs of a surgeon during a complex retinal or heart surgeries
- Prototype for a Bomb disposal robot

The screen shots of the designed robotic arm are as shown below.

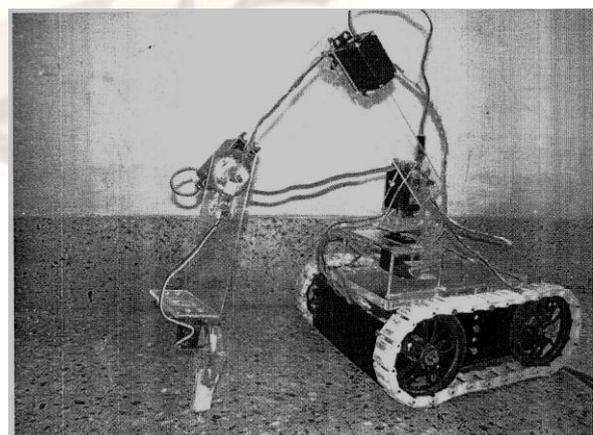


Figure 10(a) Haptic Robotic ARM Lifting Objects of average weight

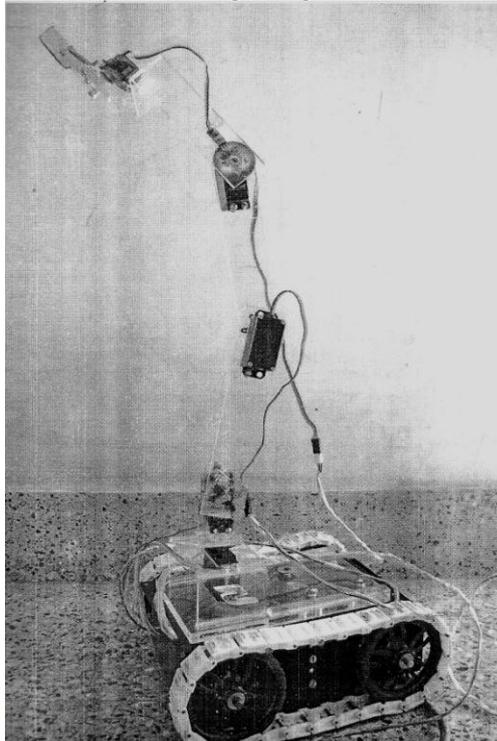


Figure 10(b) Haptic Robotic ARM

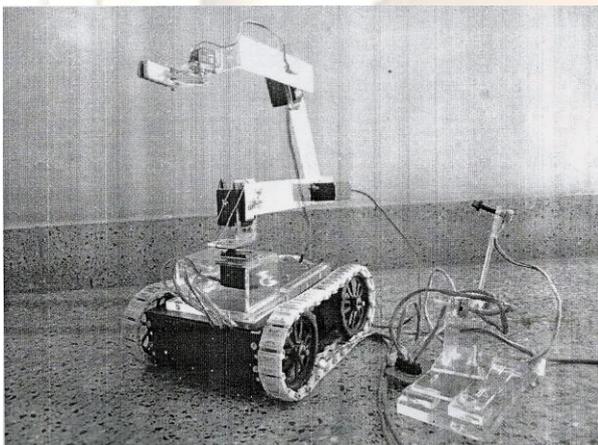


Figure 10(c) Haptic Robotic Arm cross view

It is clearly shown that the robotic arm is designed very efficiently and that the designed robotic arm is capable to lift the objects of medium weight.

FUTURE SCOPE

The robotic arm so far designed is able to lift the objects. It is able to lift the objects of medium weight. In order to extend it to some extent, more advanced tools and material with the capacity to withhold the heavy weight objects are to be used, which is then applicable in warfront and used as a rescuer at several places where there is a need and also in industrial areas, military, and so on.

REFERENCES

- [1]. Jegede Olawale, Awodele Oludele, Ajayi Ayodele, "Development of a Microcontroller Based Robotic Arm", in Proceedings of the 2007 Computer Science and IT Education Conference pg: 549-557.
- [2]. "Six-servo Robot Arm" from www.arexx.com.cn on 13-11-2011 pg: 1-18
- [3]. "Robotic Arm" - [www. NASA explores. com](http://www.nasaexplores.com) from Teacher Sheets pg:1-2
- [4]. "Robot software" from Wikipedia, the free encyclopedia on 04-3-2012
- [5]. "ATMEL – short notes", www.atmel.com
- [6]. "Arduino MCU", - www.arduino.com
- [7]. "Servomotors & system design com-ponents", - www.bhashaelectronics.com

CONCLUSION

This paper has undergone various aspects to design a robotic arm based on the haptic technology considering various aspects of it, and the basics of machine designing are observed that are explained clearly. These robots have a wide range of industrial and medical applications such as pick and place robots, surgical robots etc. They can be employed in places where precision and accuracy are required. Robots can also be employed where human hand cannot penetrate. The screen shot shows the designed robot and its functionality.