Design and fabrication of Theo Jansen's eight legged walking Robot

Papatla Rajesh[1], Rega Rajendra[2], Gangadhara Rao P[3]

[1]Phd scholar, Department of mechanical engineering, OUCE, Telangana, India
[2]Professor, Department of mechanical engineering OUCE, Telangana, India
[3]Professor, Department of mechanical Engineering Aditya College of Engineering & Technology, Surampalem, A.P, India

ABSTRACT

Since the wheel was invented back in the Stone Age, it was the primary component used in all forms of mechanical transportation. Even today it is the component of choice for almost any type of moving machine like cars. However, the wheel has always had a major disadvantage with short instant elevation changes like stairs. For most uses, climbing stairs or steep jagged rock piles is not a problem which is why the wheel is still almost always used. For the other applications, people looked at animal and human legs which are already proven to work effectively on this type of terrain. The two most effective leg mechanisms are currently Joe Klan's mechanism which resembles a spider leg and Theo Jansen's mechanism which resembles a human leg. We have chosen Theo Jansen's mechanism which has more advantage than Klan's mechanism. The main objective of our project is to replace the function of wheel with an alternative in order to overcome the difficulty of travelling in uneven terrain. This project is useful in hazardous material handling, clearing minefields or secures an area without putting anyone at risk, walking in slant positions, walking in mud without struck.

INTRODUCTION

Transporter vehicles have traditionally used wheel mechanisms like cars and trains. Wheels are ideally suited for movement without vertical fluctuations of the body, and tires with inner rubber tubes absorb shock from a rugged road. On the other hand, biologically-inspired robotics learn mobile flexibility from the morphology of multiple legs and their coordination [1]. Good examples of this are arthropods, like spiders, and the robots are conventionally designed with actuators placed in every joint. In such implementation, robots are good tools to investigate how an animal moves, but they are unable to be a substitute principle for wheels because they don‘t much take into account the maximum load capacity [2]. Joint‘s actuators promise mobile flexibility, while the actuator‘s torque performance impacts on the toughness of the robot‘s body.

Therefore, in the design of disaster robots, which need to move on rubble and carry rescue devices, continuous tracks or crawlers are popular [3].

Theo Jensen, a Dutch kinetic artist who has attempted to create a bridge between art and engineering by focusing on biological nature, proposed a linkage mechanism to mimic the skeleton of animal legs. This is called ―Theo Jansen mechanism,‖ and provides the animal with a means of moving in a fluid manner. Interestingly, his artificial animals require no electric power for actuators, and do work by weak wind power to drive the gates of multiple legs through a transformation of internal cyclic motion to an elliptical orbit of the legs [4-6].

Even in a version where the body was heavy and five meters in height, the linkage mechanism worked smoothly for walking with minimal power loss. Concepts of the linkage mechanism are in fact found extensively in heavy industrial machines, which are accompanied with hydraulic actuators, such as cranes and shovels. Thus, the linkage has the potential to act as a substitute for wheels, especially in rugged fields. A problem with the Theo Jansen mechanism is the availability and extensibility of walking patterns under bumpy conditions.

The current mechanism concentrates on smoothness at the precise moment the legs touch the ground, and minimizes the force of impact in the toe in order to prevent vertical fluctuations of the body and the breakage parts in the case that the body is heavy. It brings weakness in adaptation to changes of walking fields especially in the presence of obstacles. In order to lift the legs during locomotion, an extension mechanism is crucial for transitions between walking and climbing modes. We investigate the possibility of designing a Theo Jansen mechanism involving multiple modes and the generation of a continuous orbit of the legs connecting them. Finally, we propose a solution by using an additional up-and-down motion placed at the joint center [7-10]

PROPOSED SYSTEM

Block diagram:

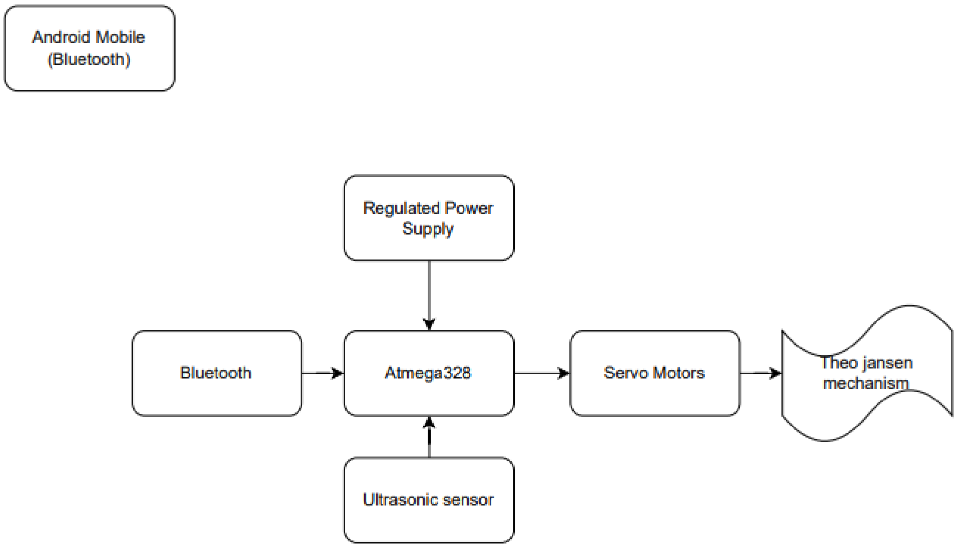


Fig.1 Block Diagram

As mentioned earlier we intend to design a replacement for tyres in load carrying trucks in mineral excavation belts which face a lot of wear and tear due to poor haul road construction. To make the design possible I needed reference data. Using the information power of the internet I came across a report journal on —Finite Element Stress analysis of a solid tire‖.

Issue 2 December 2008). The report involves the standard Finite element analysis of a solid tire and deflections produced in it on application of load. The detail related to the analysis is given further in the report. With reference to the Journal report I decided the dimensions of our setup. The easy aspect of Theo-Jansen Mechanism (STRANDBEEST) is that it has been provided with a specific set of dimensions which can be scaled to any level, bet it on a small scale as in toys or an industrial scale, i.e. my idea

As we show above, these dimension sets are standard and can be used as a scaling reference. During the course of the research it was also found during rough designing that irrespective of the length of the linkages involved the system worked perfectly and the only drawback faced was low stride height.

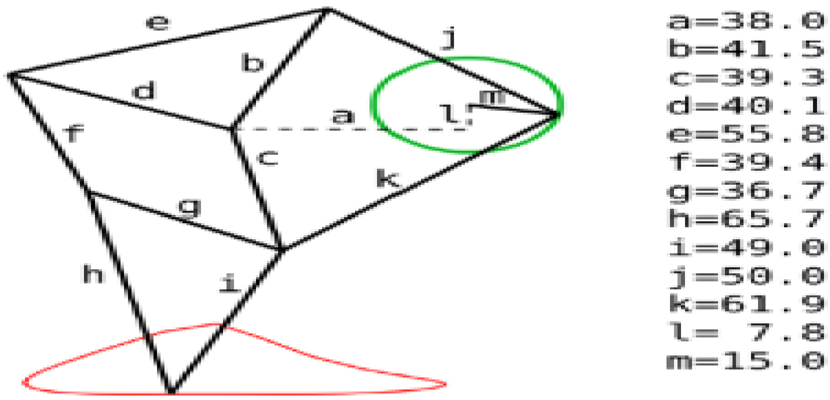


Fig. 2Dimensions scale taken as reference

Fig.2 Dimensions Scale

Before proceeding further we need to explain about the stride pattern in walking mechanisms. Every linkage based walking mechanism either autonomous or manually operated follows a certain pattern while moving. These patterns can be represented as imaginary geometric figures which can be further studied to improve the walking and make it smoother. The more edgy the stride pattern the more jerky the motion of the mechanism. A lot of online reports have been prepared where students from different universities have come up with ideas to replace wheels in cycles and also mechanism for bearing heavy loads and can be moved smoothly. With reference to these projects I referred to the stride pattern of the Jansen mechanism.

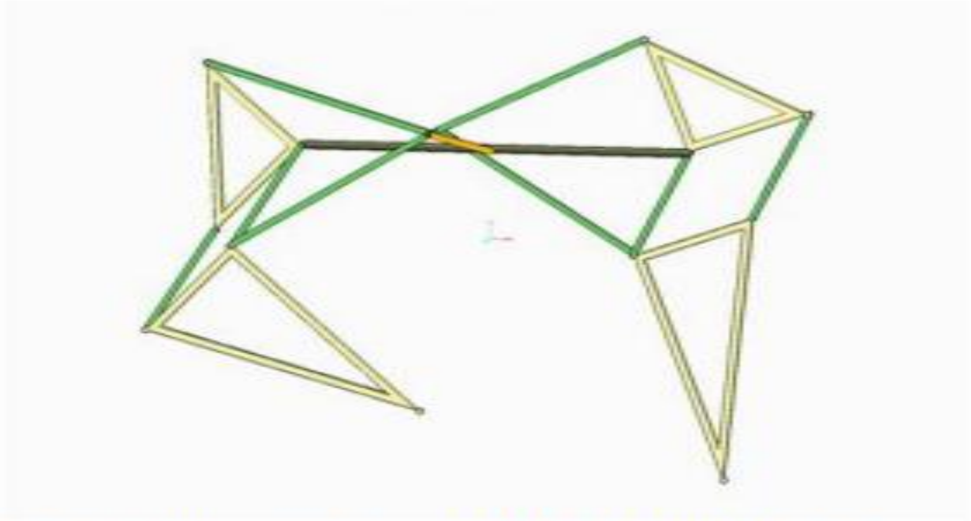


Fig. 4Software Designed Model (Auto CAD)

Fig.3 Software Design

With these references for help I started to develop a design for the system considering dimension in comparison to the tyre analysis report. Initially I came up with a rough line diagram of the design as seen from the right-side view and depicted its motion pattern segregated into three unique diagrams. The following gives a brief description of the motion of the vehicle and its relative placement under the chassis of the truck. The below figures give a brief idea of how the system might work. On considering a practical model of the above line diagram, we intend to replace the front and rear axle along with the tyres with this unique mechanism where in each axle both front and rear will have 4 sister pairs. Thus, we would have 8 pairs in total and overall all 16 legs to support the system. From our survey related to mining and transport we found that a standard truck while loaded has to bear a load up to 16,000 kg. Keeping that in mind I intend to design the system to bear static load of 16,000 kg and accordingly carry out compressive stress analysis across its T joint sections while the body is in static condition. Taking into consideration the analysis report of the tyre I found that the dimensions of a standard tire. The tire model has an outer diameter of 518 mm, 218mm inner diameter and a tyre width of 144 mm. On further calculation, I found the standard dimensions of the mechanism has to scaled 4 times its original and that is how I came up with the standard model which I drew using NXCAD. The following depicts a brief model of on unit, i.e. two sister pairs.

Table.1 Na,E Details Table

Pin Category	Pin Name	Details
Power	**Vin, 3.3V, 5V, GND**	**Vin:** Input voltage to Arduino when using an external power source (6-12V). **5V:** Regulated power supply used to power microcontroller and other components on the board. **3.3V:** 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. **GND:** Ground pins.
Reset	**Reset**	Resets the microcontroller.
Analog Pins	**A0 – A7**	Used to measure analog voltage in the range of 0-5V
Input/Output Pins	**Digital Pins D0 - D13**	Can be used as input or output pins. 0V (low) and 5V (high)
Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.

External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	**13**	To turn on the inbuilt LED.
IIC	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	**AREF**	To provide a reference voltage for input voltage.

HARDWARE DESCRIPTION

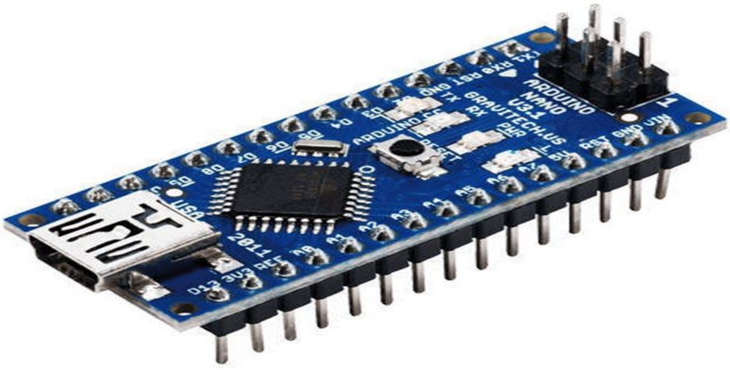


Fig.4 Arduino Nano

Microcontroller	ATmega328P – 8-bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage for Vin pin	7-12V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (2 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Communication	IIC, SPI, USART
TABLE 2:MICROCONTROLLER	

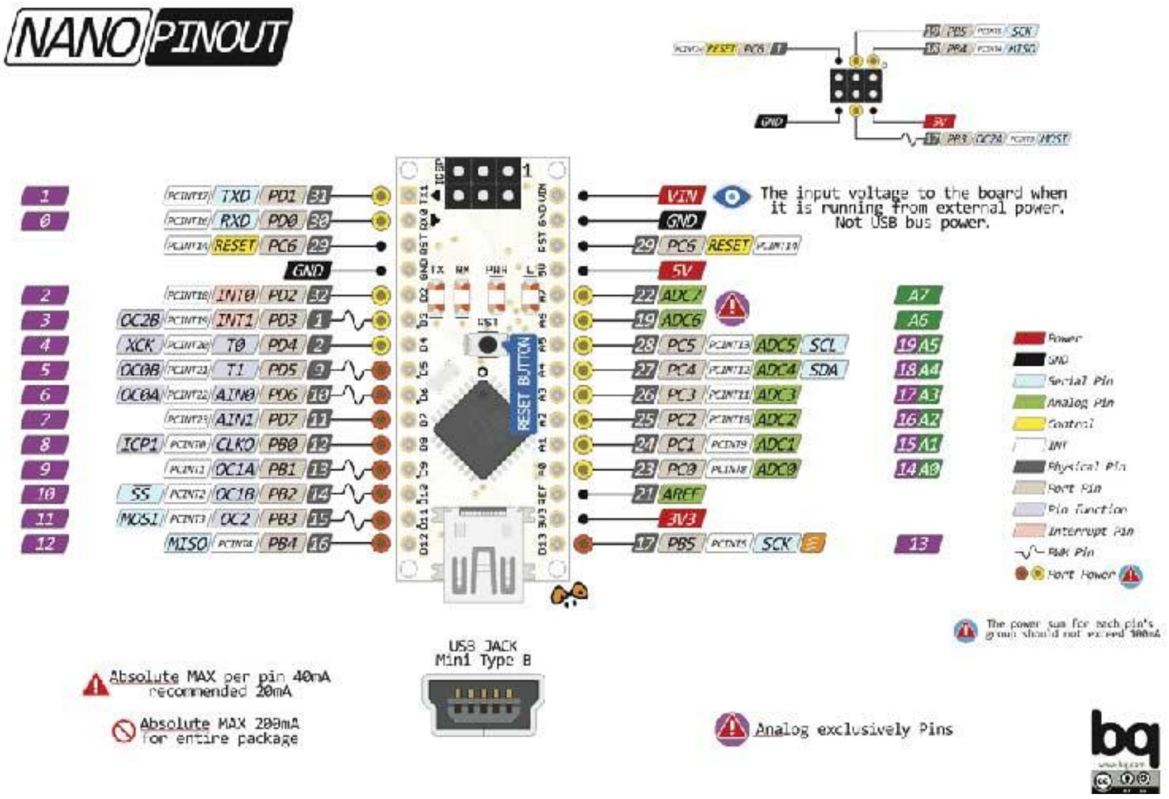


Fig.5 Arduino Nano Development Board Arduino Nano Pin Diagram

The **Arduino Nano** is another popular Arduino development board very much similar to the Arduino UNO. They use the same Processor (Atmega328p) and hence they both can share the same program.

Arduino Nano Pinout Configuration
Arduino Nano Technical Specifications
Difference between Arduino UNO and Arduino Nano

The **Arduino Nano** is very much similar to the Arduino UNO. They use the same Processor (Atmega328p) and hence they both can share the same program. One big difference between both is the size. UNO is twice as big as Nano and hence occupies more space on your project. Also, Nano is breadboard friendly while Uno is not. To program an Uno, you need a Regular USB cable; whereas for Nano, you will need a mini USB cable. The **technical difference between Uno and Nano** is shown below:

Table 3: Microprocessor

Name	**Processor**	**Operating/Input Voltage**	**CPU speed**	**Analog In/Out**	**Digital IO/PWM**	**EEPROM / SRAM[kB]**	**Flash**	**USB**	**USART**
Uno	ATmega328P	5V / 7-12V	16 MHz	**6 / 0**	14 / 6	1 / 2	32	**Regular**	1
Nano	ATmega328P	5V / 7-12V	16 MHz	**8 / 0**	14 / 6	1 / 2	32	**Mini**	1

ESP32-CAM

The ESP32-CAM is a development board with an ESP32-S chip, an OV2640 camera, microSD card slot and several GPIOs to connect peripherals. In this guide, we'll take a look at the ESP32-CAM GPIOs and how to use them.

Pinout Diagram

The following image shows the pinout diagram for the ESP32-CAM AI-Thinker.

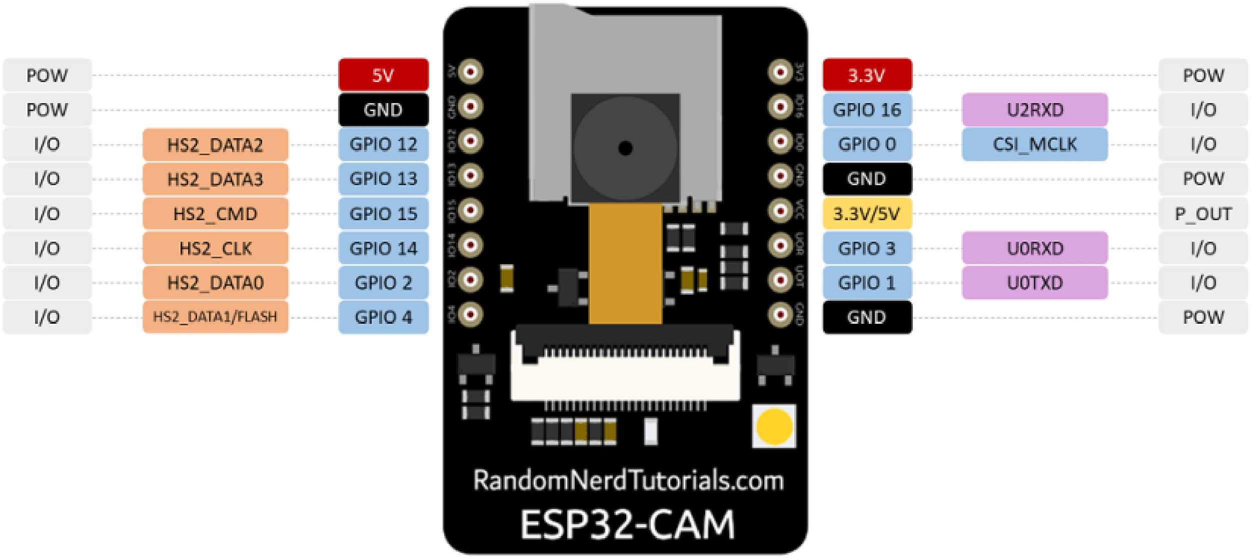


Fig.6 Pinout Diagram

Power Pins

The ESP32-CAM comes with three GND pins (colored in black color) and two power pins (colored with red color): 3.3V and 5V.

You can power the ESP32-CAM through the 3.3V or 5V pins. However, many people reported errors when powering the ESP32-CAM with 3.3V, so we always advise to **power the ESP32-CAM through the 5V pin**.
Power output pin

There’s also the pin labeled on the silkscreen as **VCC** (colored with a yellow rectangle). You should not use that pin to power the ESP32-CAM. That is an output power pin. It can either output 5V or 3.3V.

In our case, the ESP32-CAM outputs 3.3V whether it is powered with 5V or 3.3V. Next to the VCC pin, there are two pads. One labeled as 3.3V and other as 5V.

ULTRASONIC SENSOR

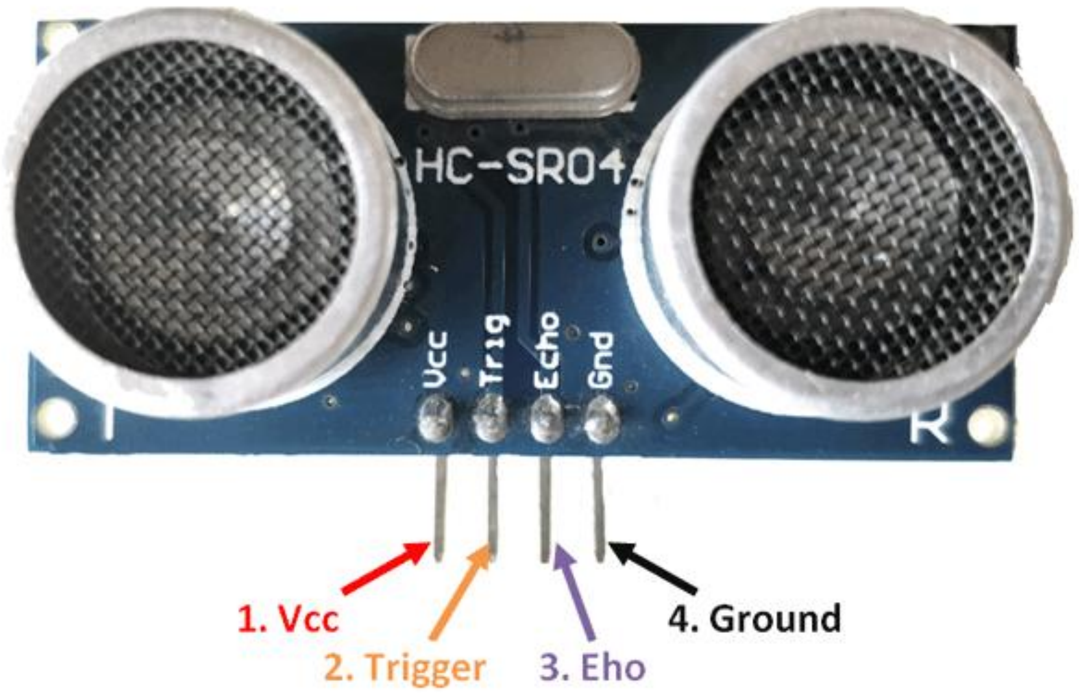


Fig.7 Ultrasonic Sensor

Ultrasonic Sensor HC SR04
Ultrasonic Sensor HC SR04 Pin Diagram

Ultrasonic Sensor Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

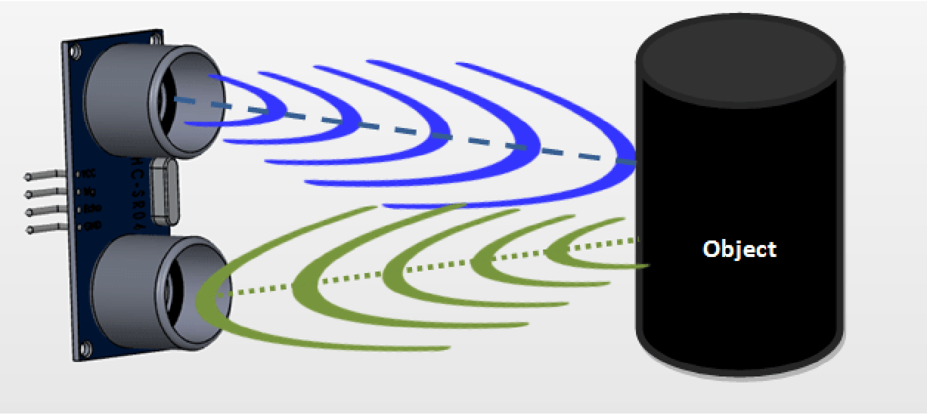


Fig.8 Working Of Ultrasonic Sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

BLUETOOTH MODULE (HC-05):
Overview: Communication device:-over project is based on wireless communication between micro controller and mobile phone. But alone micro controller is not able to communicate directly to the android mobile phone. Bluetooth Serial module’s operation doesn’t need drive, and can communicate with the otherBluetooth device that has the serial. But communication between two Bluetooth modules requires at

Least two conditions:
(1) The communication must be between master and slave.
(2) The password must be correct.

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module,designed for transparent wireless serial connection setup.Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (EnhancedData Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver andbaseband. It uses CSR Blue core 04-External single chip Bluetooth system withCMOS technology and with AFH (Adaptive Frequency Hopping Feature). Ithas the Footprint as small as 12.7mm x 27mm.

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband Bluetooth Wireless networks for short range communications have a wide spread usage of Bluetooth radio transmissions between 2400–2480 MHz Modern mobile devices embed small, low-powered and cheap integrated chips functioning as short-range radio transceivers for Bluetooth radio communications. Device pairing, authentication, encryption and authorization techniques have given recognition to Bluetooth technology due to its vital security mechanisms.

Different types of Bluetooth applications can be developed using Android platform architecture using the Bluetooth profiles. The device manufacturers provide the services using the support of these profiles in their devices to maintain compatibility for the Bluetooth technology

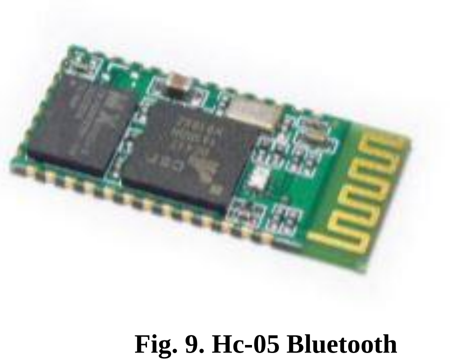


Fig. 9. Hc-05 Bluetooth

Specifications
Hardware features

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmits power.
- Low Power 1.8V Operation, 3.3 to 5 V I/O.
- PIO control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

Software features

- Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- PIO9 and PIO8 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto connects to the last device on power as default.
- Permit pairing device to connect as default.
- Auto pairing PINCODE:"1234" as default.
- Auto reconnect in 30 min when disconnected as a result of beyond the range of connection.

Pin out configuration

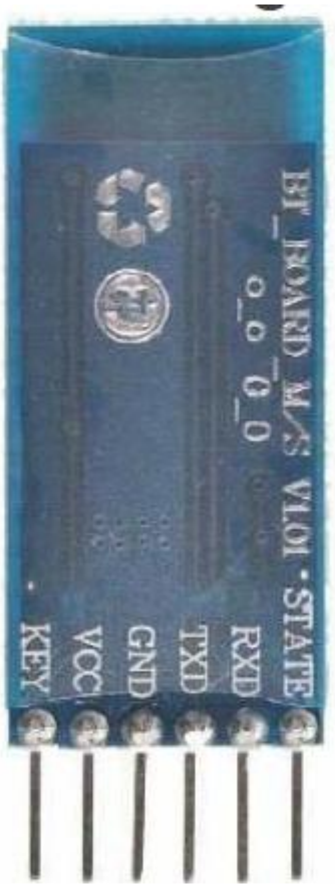


FIG. 10: PIN-OUT OF HC-05

Typical Application Circuit:

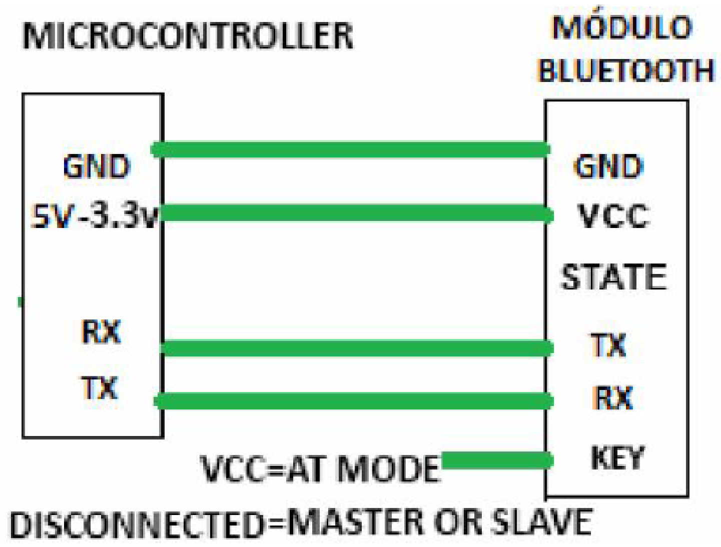


Fig. 11: Vcc Circuit

Pairing:
After connect the Bluetooth module, scan for new devices from the Android phone and you will find the module with the device name “HC-05”, after that, click to connect, if some message appears asking about “Pairing code” just put **“1234”**as default code.
BLUE LED = ACTIVE (Blinking 500ms period inactive connection, change 1seg with active connection)

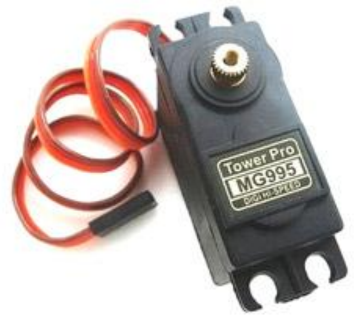
How to get to the standard communication mode

1. Leave free KEY, don’t connect it to VDD neither GND.
2. Supply power to the module. Then the module will enter tocommunication mode. It can be used for pairing.

HC-05 BLUETOOTH MODULE WORKING VOLTAGE:-
The Bluetooth module HC-05 is used to receive & transmit data between Bluetooth device and MCU. It requires power supply from 3.3V to 5V.

SERIAL COMMUNICATION:-
To transfer to a device located many meters away, the serial method is used.The data is sent one bit at a time. Here not 8bit data is send 2 extra bit are send along with it .this two bit are called start bit and stop bit. These tow bit are used so synchronization can be done between transmitter and receiver.

MG995 Servo Motor
20 November 2020 - 0 Comments



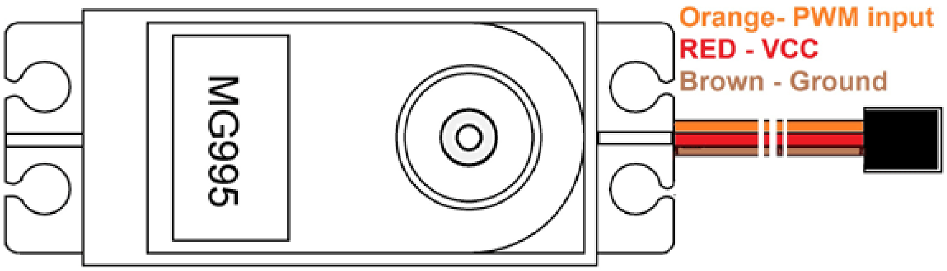


Fig. 12: Servo Motor

MG995 is a **servo motor** that is popular for its performance and low price. The motor is used in many applications mainly being robotics and drones.

MG995 Servo Motor Pinout Configuration
MG995 has three terminals as mentioned in pin diagram and the function of each pin is given below.

Pin	Name	Function
1	Signal pin (Orange pin)	The PWM signal which states the axis position is given through this pin.
2	VCC (Red pin)	Positive power supply for servo motor is given to this pin.
3	Ground(Brown pin)	This pin is connected to ground of circuit or power supply.

SOFTWARE DESCRIPTION

ARDUINO IDE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

✓ Verify
Checks your code for errors compiling it.

➜ Upload
Compiles your code and uploads it to the configured board. See uploading below for details.
Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

🗎 New
Creates a new sketch.

⬆ Open
Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.
Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the **File | Sketchbook**menu instead.

⬇ Save
Saves your sketch.

🔍 SerialMonitor
Opens the serial monitor.

Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, **Help**. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- New
  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open
  Allows to load a sketch file browsing through the computer drives and folders.
- OpenRecent
  Provides a short list of the most recent sketches, ready to be opened.
- Sketchbook
  Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- Examples
  Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- Close
  Closes the instance of the Arduino Software from which it is clicked.

- Save
  Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- Saveas...
  Allows to save the current sketch with a different name.
- Page Setup
  It shows the Page Setup window for printing.
- Print
  Sends the current sketch to the printer according to the settings defined in Page Setup.
- Preferences
  Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- Quit
  Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.
  Edit
- Undo/Redo
  Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut
  Removes the selected text from the editor and places it into the clipboard.
- Copy
  Duplicates the selected text in the editor and places it into the clipboard.
- Copy for Forum
  Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- Copy as HTML
  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste
  Puts the contents of the clipboard at the cursor position, in the editor.
- Select All
  Selects and highlights the whole content of the editor.
- Comment/Uncomment
  Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent
  Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find
  Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next
  Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous
  Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.
  Sketch
- Verify/Compile
  Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload
  Compiles and loads the binary file onto the configured board through the configured Port.
- Upload Using Programmer
  This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.

- Export Compiled Binary
  Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder
  Opens the current sketch folder.
- Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

- Add File...
  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

Tools

- Auto Format
  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- Archive Sketch
  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- Fix Encoding & Reload
  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial Monitor
  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board
  Select the board that you're using. See below for descriptions of the various boards.
- Port
  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer
  For selecting a harware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- Burn Bootloader
- The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the **Boards** menu before burning the bootloader on the target board. This command also set the right fuses.

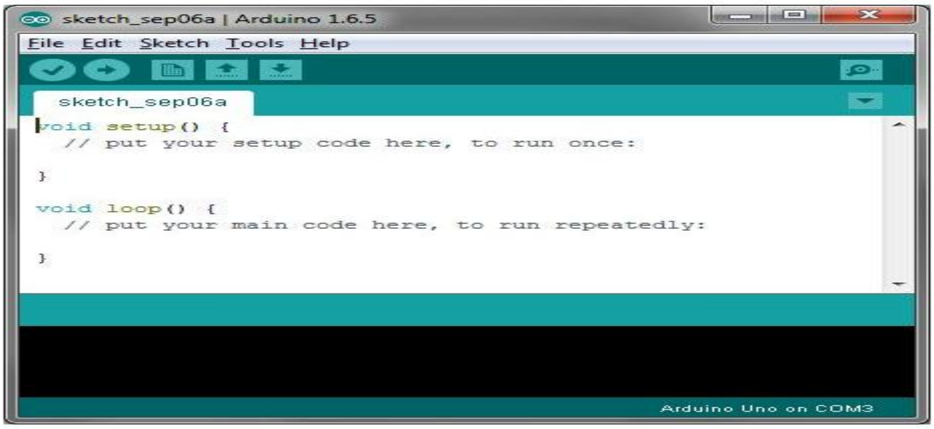
Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

- Find in Reference
  This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Arduino IDE: Initial Setup

This is the Arduino IDE once it's been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

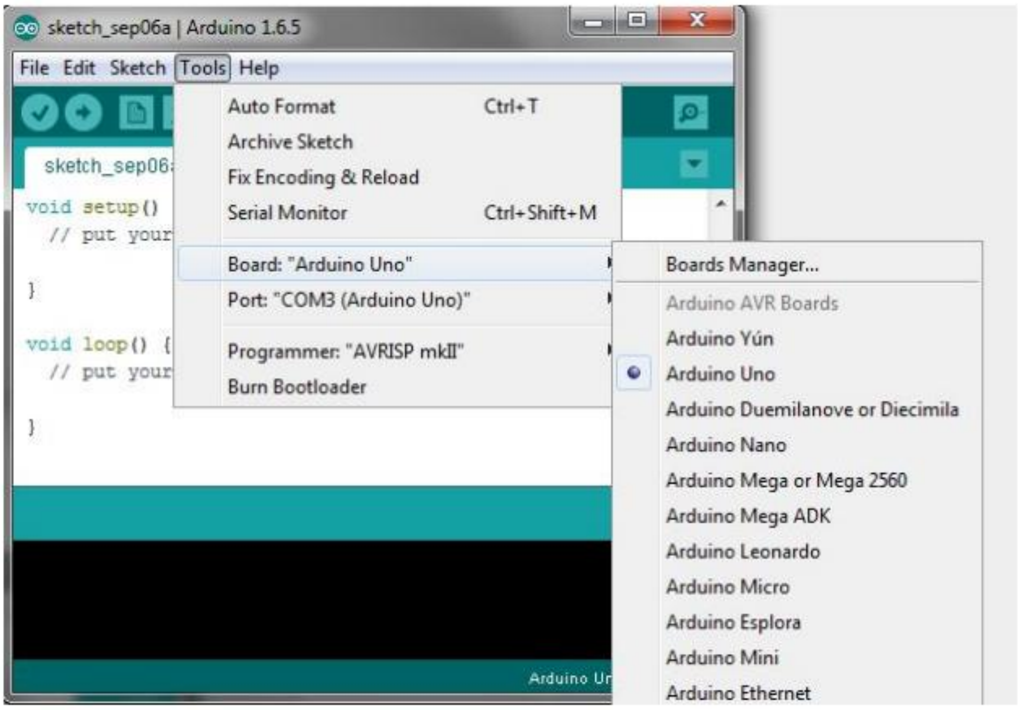
Arduino Uno on COM3

Arduino IDE Default Window

Fig. 13: Arduino Ide Window

IDE: Board Setup

You have to tell the Arduino IDE what board you are uploading to. Select the Toolspulldown menu and go to Board.This list is populated by default with the currently available Arduino Boards that are developed by Arduino. If you are using an Uno or an Uno-Compatible Clone (ex. Funduino, SainSmart, IEIK, etc.), select Arduino Uno. If you are using another board/clone, select that board.

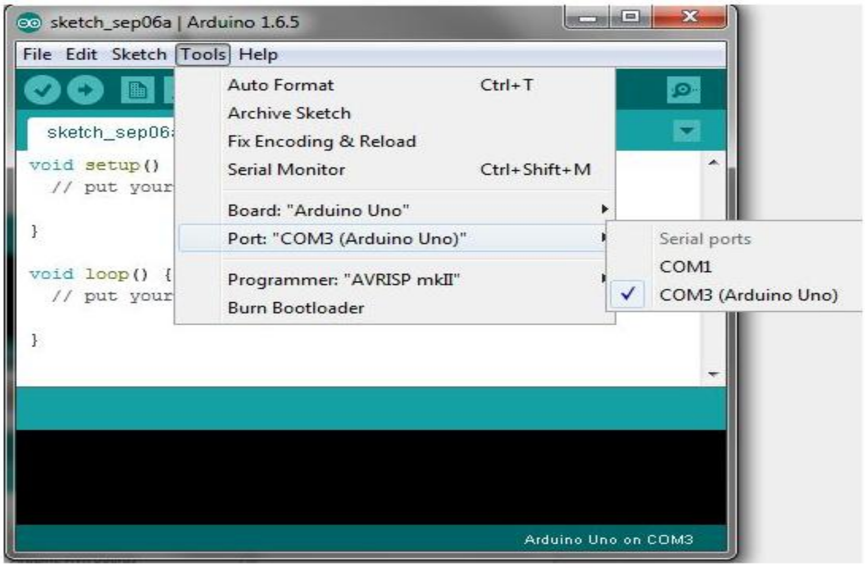


Arduino IDE: Board Setup Procedure

Fig. 13: Procedure

IDE: COM Port Setup

If you downloaded the Arduino IDE before plugging in your Arduino board, when you plugged in the board, the USB drivers should have installed automatically. The most recent Arduino IDE should recognize connected boards and label them with which COM port they are using. Select the Tools pulldown menu and then Port.Here it should list all open COM ports, and if there is a recognized Arduino Board, it will also give it's name. Select the Arduino board that you have connected to the PC. If the setup was successful, in the bottom right of the Arduino IDE, you should see the board type and COM number of the board you plan to program. Note: the Arduino Uno occupies the next available COM port; it will not always be COM3.

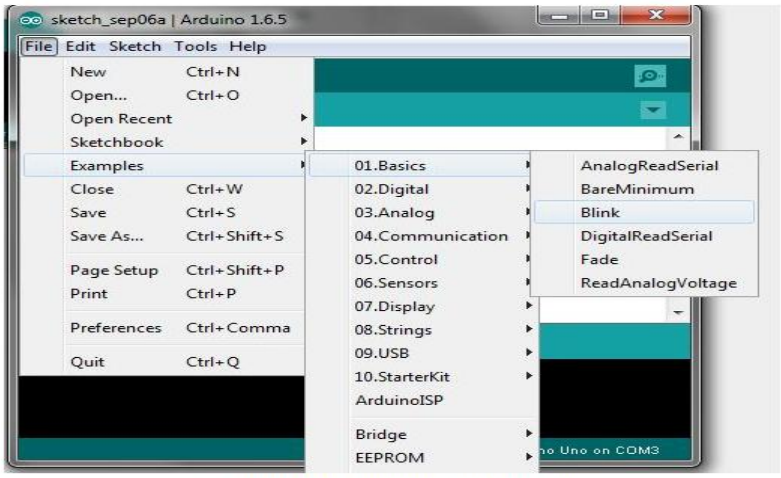


Arduino IDE: COM Port Setup

Fig. 14: Com Port Setup

Testing Your Settings:
Uploading Blink One common procedure to test whether the board you are using is properly set up is to upload the "Blink" sketch. This sketch is included with all Arduino IDE releases and can be accessed by the Filepull-down menu and going to Examples, 01.Basics, and then select Blink. Standard Arduino Boards include a surface-mounted LED labeled "L" or "LED" next to the "RX" and "TX" LEDs, that is connected to digital pin 13. This sketch will blink the LED at a regular interval, and is an easy way to confirm if your board is set up properly and you were successful in uploading code. Open the "Blink" sketch and press the "Upload" button in the upper-left corner to upload "Blink" to the board.

Upload Button:



Arduino IDE: Loading Blink Sketch

Fig. 15: Testing

RESULTS AND DISCUSSION

Final result is the assembly of product, and it is shown in figure. This mechanism can walk forward and back word. Also this robot can be operated by using computer program. The strength of each link is just enough to carry self weight as well as machine control unit. The nuts and bolts and axils used in this mechanism are made of stainless steel. It can be mentioned as four legs front and four legs back. All legs will move with sequential manner so that there is a continuous motion, with minimum or almost zero acceleration and deceleration accept during starting and stopping situations.

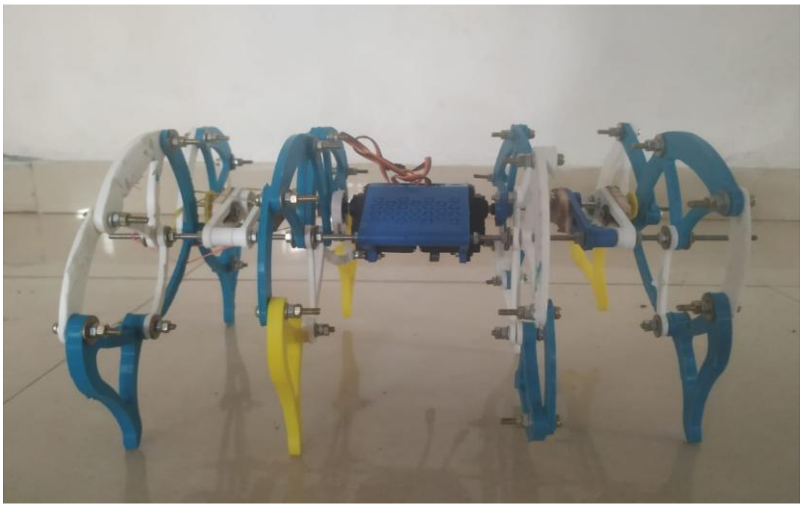


Fig. 15: Final Prototype Of Model

CONCLUSIONS

As for the actual walker mechanism, much was learned about Theo Jansen devices during the course of this project, including joint quality, required number of linkage sets for stability, weight considerations and driven power availability, etc. With all the knowledge and techniques acquired from the project, we were confident about designing and fabricating a greatly improved Theo Jansen device within a shorter time.

REFFERENCES

[1]. Liu, Weiping, and LinyuanLü. "Link prediction based on local random walk." EPL (Europhysics Letters) 89.5 (2010): 58007.

[2]. Harris, N. G. "Modelling walk link congestion and the prioritisation of congestion relief." Traffic Engineering & Control 32.2 (1991).

[3]. Link, Tanja, et al. "Walk the number line–An embodied training of numerical concepts." Trends in Neuroscience and Education 2.2 (2013): 74-84.

[4]. Hemami, H., Yuan-Fang Zheng, and M. J. Hines. "Initiation of walk and tiptoe of a planar nine-link biped." Mathematical Biosciences 61.2 (1982): 163- 189.

[5]. Koozekanani, S. H., et al. "On the role of dynamic models in quantitative posturography." IEEE Transactions on biomedical engineering 10 (1980): 605-609.

[6]. Collins-Thompson, Kevyn, and Jamie Callan. "Query expansion using random walk models." Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005. [6] Li, Rong-Hua, Jeffrey Xu Yu, and Jianquan Liu. "Link prediction: the power of maximal entropy random walk." Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, 2011.

[7]. Likhtman, Alexei E. "Single-chain slip-link model of entangled polymers: Simultaneous description of neutron spin− echo, rheology, and diffusion." Macromolecules 38.14 (2005): 6128-6139.

[8]. Romanelli, A., et al. "Decoherence in the quantum walk on the line." Physica A: Statistical Mechanics and its Applications 347 (2005): 137-152.

[9]. Zheng, Lianqing, Mengen Chen, and Wei Yang. "Random walk in orthogonal space to achieve efficient free-energy simulation of complex systems." Proceedings of the National Academy of Sciences 105.51 (2008): 20227-20232.

[10]. K. Srinivas, A. Lakshumu Naidu, and M. V. A. RajuBahubalendruni. "A Review on Chemical and Mechanical Properties of Natural Fiber Reinforced Polymer Composites." International Journal of Performability Engineering 13.No. 2 (2017): 189-200