GESTURE CONTROLLED
WIRELESS AGRICULTURAL WEEDING ROBOT

A PROJECT REPORT

Submitted by

THARUN KUMAR C [211419104293]

VISHNUDHAR G R [211419104309]

RAMAKRISHNA P [211419104003]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report **"GESTURE CONTROLLED WIRELESS AGRICULTURAL WEEDING ROBOT"** is the Bonafide work of "**THARUN KUMAR C [211419104293], VISHNUDHAR G R [211419104309] ,RAMAKRISHNA P [211419104230] "** who carried out the project work under my supervision.

SIGNATURE **SIGNATURE**

Dr .L. JABASHEELA, M.E., Ph.D.,
PROFESSOR
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Dr.M.MOHAN, B.E,M.Tech,Ph.D
SUPERVISOR
ASSISTANT PROFESSOR
GRADE 1

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
COLLEGE, NASARATHPETTAI
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the End Semester Viva-Voce Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **THARUN KUMAR C [REG.NO: 211419104293], VISHNUDHAR G R [REG.NO:211419104309], RAMAKRISHNA P [REG.NO:211419104230]** hereby declare that this project report titled **"GESTURE CONTROLLED WIRELESS AGRICULTURAL WEEDING ROBOT" ,** under the guidance of Dr.K.MOHAN is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

<< SIGNATURE OF ALL STUDENTS IN THE BATCH>>

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide, **Dr. M.MOHAN, B.E, M.Tech,Ph.D,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

NAME OF THE STUDENTS

THARUN KUMAR C

VISHNUDHAR G R

RAMAKRISHNA P

ABSTRACT

Automated processes in the field of agriculture have become more and more reliable and efficient. There are many difficulties faced when manpower is used. It is time. consuming and becomes tedious. Robotic systems integrated with various control. methods can be very useful in doing repetitive work, such as seed sowing process. where the same movement is continuous. Previous weed removal robots included. optical image sensing which makes the system costlier. Our robot is cost-effective. which eliminates optical sensors. In this project, we have developed a Trainable automatic robot which helps in removing unwanted weed on agricultural fields using gesture to control a three-axis robotic arm to do the necessary work. The arm is placed. on a rover which is controlled Wirelessly using Bluetooth. The arm is taught to. do the repetitive motion with a gesture using a hand glove to do the necessary work. The setup of the rover with attached the robotic arm is tested and evaluation under normal environmental conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In the field of agriculture, various operations for handling heavy material are performed. For example, in vegetable cropping, workers should handle heavy vegetables in the harvest season. Additionally, in organic farming, which is fast gaining popularity, workers should handle heavy compost bags in the fertilizing season. These operations are dull, repetitive, or require strength and skill for the workers. In the 1980‟s many agricultural robots were started for research and development. Kawamura and co-workers developed the fruit harvesting in orchard. Grand and co-workers developed the apple harvesting robot. They have been followed by many other works.

Many of the works focus on structure systems design (e.g., mechanical systems design) of robot and report realization of the basic actions in actual open fields. However, many of the robots are not in the stages of diffusion but still in the stages of research and development. It is important to find rooms to achieve higher performance and lower cost of the robots. Agriculture involves the systematic production of food, feed, fibre, and other goods. In addition to producing food for humans and animals, agriculture also produces cut flowers, timber, fertilizers, animal hides, leather, and industrial chemicals.

CHAPTER 2

LITERATURE REVIEWS

CHAPTER 2
LITERATURE REVIEWS

[1] Title Detection and Get Rid of Blockage in a Manhole Pipe Using IoT

Authors Priyanka Y K

Published Year 2020

Drawbacks

- Prone to Errors
- Generally, have high polynomial running times.

Description

Internet of things (IoT) remains one of the most electrifying and fast burgeoning fields. It permits most of the things like any objects are controlled with very less manpower and we can control things anywhere and anytime by using network. In this paper that going to be constructing that the “Detection of Blockage in Manhole Using IoT”. Detecting a struck in the manhole pipe, due to blockage of sludge in a pipe. Here by using detecting the length of the pipe using ultrasonic sensor till where the sludge struck with the help of capturing image of sludge to aware of, due to gloominess inside a manhole by provide a luminescence, which helps to capture picture of sludge. And with the help of developing an application to control the device, by using that application data access can be done (captured picture) using a cloud. In traditional method, sewers must do cleaning in all the cities (areas). In

gutters, workers stand in the waste, which reaches chest level and with the help of long wooden sticks and broomsticks they clear blockage. In some area's workers crawl into a sewage, without wearing any protective gear. And those workers are treated very badly by the society. Thus, sewers need to work a lot to clear blockage. But now by using this technique it will be very much easier to reduce a manpower and also incremental steps can be added further.

[2] Title Computational Acoustic Model for Non-intrusive Inspection of a Fluidic

Authors Oluwatosin Ogundare, Srinivasan

Published Year 2020

Drawbacks

- Difficult to be used in large-scale parallel computing.
- High complexity of installing and maintaining

Description

The shortcomings of traditional pressure wave analysis for the detection of material deposits and structural compromises within a pipeline forms the motivation for this work. In many Oil and Gas pipelines, a pressure pulse generated using a fast-acting valve (hydrodynamic pressure) or an intrusive acoustic source (acoustic pressure) is often used for leak detection, deposition or blockage detection. A data logger records the incident pressure wave and its reflection. The mechanism of wave generation sometimes limits the usefulness of pressure wave analysis for near field pipeline diagnostics. In this regard, acoustic reflectometry performs well for near field analysis. However, the implicit requirement of an intrusive acoustic source limits mainstream adoption in pipelines. To mitigate this problem, a non-intrusive approach to acoustic wave analysis in a pipeline is introduced.

[3] Title A Novel Step Optimal Path Planning Algorithm for the Spherical Mobile Robot Based on Fuzzy Control

Authors JIAN GUO 1 , (Fellow, IEEE), CHUNYING LI 1 , (Student Member, IEEE), AND SHUXIANG GUO

Published Year 2019

Drawbacks

- Generally, have high polynomial running times.
- Unsuitable for large scale scenarios

Description

In order to improve the ability of the spherical mobile robot to navigate and move autonomously in an unknown environment. This paper proposed a novel step optimal path planning method based on fuzzy control. Firstly, by analyzing the motion model of the spherical mobile robot, the arrangement and debugging of the ultrasonic sensors (HC-SR04) were completed. Then, through multi-sensor fusion technology and D-H parameter method, a fuzzy controller for the spherical mobile robot was designed. Finally, the proposed fuzzy control method was applied to the path planning of spherical mobile robot in unknown environment and tested by a series of simulations and experiments. The results showed that the step optimal method based on fuzzy control could ensure that the spherical mobile robot completed the path planning. The experimental results also verified the effectiveness and practicability of the proposed novel path planning method.

[4] Title An Open-Source Low-Cost Mobile Robot System With an RGB-D Camera and Efficient Real-Time Navigation Algorithm

Authors TAEKYUNG KIM 1, SEUNGHYUN LIM 2, (Graduate Student Member, IEEE), GWANJUN SHIN

Published Year 2022

Drawbacks

- Complexity of its Real Time Implementation

- Tedious message updating

Description

Currently, mobile robots are developing rapidly and are finding numerous applications in the industry. However, several problems remain related to their practical use, such as the need for expensive hardware and high-power consumption levels. In this study, we build a low-cost indoor mobile robot platform that does not include a LiDAR or a GPU. Then, we design an autonomous navigation architecture that guarantees real-time performance on our platform with an RGB-D camera and a low-end off-the-shelf single board computer. The overall system includes SLAM, global path planning, ground segmentation, and motion planning. The proposed ground segmentation approach extracts a travers ability map from raw depth images for the safe driving of low-body mobile robots. We apply both rule-based and learning-based navigation policies using the travers ability map. Running sensor data processing and other autonomous driving components simultaneously, our navigation policies perform rapidly at a refresh rate of 18 Hz for control command, whereas other systems have slower refresh rates. Our methods show better performances than current state-of-the-art navigation approaches within limited computation resources as shown in 3D simulation tests. In addition, we demonstrate the applicability of our mobile robot system through successful autonomous driving in an indoor environment.

[5] Title Energy Comparison of Controllers Used for a Differential Drive Wheeled Mobile Robot

Authors ALEXANDR STEFEK 1 , THUAN VAN PHAM 1 , VACLAV KRIVANEK 2 , (Member, IEEE), AND KHAC LAM PHAM

Published Year 2020

Drawbacks

- High complexity, inaccuracy, and inadequacy
- Generally have high polynomial running times.

Description

In order to select the best controller for a Differential Drive Wheeled Mobile Robot (DDWMR), an energy consumption comparison relating to tracking accuracy is used as a very strict criterion. Therefore, this paper reviews some well-known controllers designed for the DDWMR. Furthermore, there are presented several experiments with the extensible open-source code programmed in Python. Such an extensible open-source code presentation could serve as a tool for simulating, comparing, and evaluating a set of different control algorithms. The kinematic and dynamic models of the DDWMR and control algorithms are implemented in this open-source code to determine a travel time, a distance between the robot's position and a given path, a linear velocity, an angular velocity, a travel path length, and a total kinetic energy loss of the DDWMR. These simulation results are used to compare and evaluate the given control algorithms. Moreover, the simulation results also enable to answer the question of whether a significant increase in energy consumption is worth shortening the travel path by just a bit. Finally, this paper includes a direct link to the stored experiments which are runnable and could serve as a proof. Besides, users can also easily supplement with other controllers and different paths to evaluate robot tracking control algorithms

CHAPTER 3

SYSTEM ANALYSIS

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

There are several existing systems for agricultural robots that are being used by farmers around the world. Here are some examples:

John Deere's "See & Spray" technology: This system uses computer vision to detect and classify plants, and then applies herbicides only to the weeds, reducing the amount of chemicals needed for crop management.

Yamaha's "YSMART" system: This system uses drones to spray pesticides and fertilizers on crops, reducing the amount of chemicals needed and increasing efficiency.

Harvest Automation's "HV-100" system: This system uses small robots to move potted plants around nurseries and greenhouses, reducing the need for manual labor.

Blue River Technology's "See & Spray" system: Similar to John Deere's system, this system uses computer vision to detect and classify plants, and then applies herbicides only to the weeds.

Lely's "Discovery 90 SW" system: This system is a robotic milking system that allows cows to be milked automatically, reducing the need for manual labor and increasing efficiency.

Overall, these existing systems for agricultural robots are designed to automate specific tasks such as spraying, planting, harvesting, and milking. By using robots to perform these tasks, farmers can increase efficiency and reduce labor costs, while also reducing the amount of chemicals needed for crop management. As technology continues to advance, we can expect to see even more innovative systems for agricultural robots in the future.

DISADVANTAGES

- Limited Depth of Penetration
- Expensive
- Less accuracy

3.2 PROPOSED SYSTEM

Agricultural robots are becoming increasingly popular as they can perform various tasks such as planting, watering, harvesting, and weeding with greater precision and efficiency than human labour. Here are some key components that can be included in a proposed system for an agricultural robot:

Sensing and perception: The robot should be equipped with sensors such as cameras, lidar, and radar to detect and perceive its surroundings. This allows the robot to identify and locate crops, weeds, pests, and other obstacles in the field.

Navigation and mapping: The robot should have the ability to navigate autonomously through the farm using GPS, SLAM (Simultaneous Localization and Mapping), and other techniques. A map of the farm can be

created using the robot's sensors, and this map can be updated in real-time as the robot moves through the field.

Manipulation and actuation: The robot should be capable of manipulating objects such as plants, soil, and tools. This can be achieved through the use of robotic arms, grippers, and other mechanisms. The robot should also be able to actuate various tools and devices such as sprayers, cultivators, and harvesters.

Data analysis and decision-making: The robot should be able to analyze data from its sensors and make decisions based on this data. For example, the robot can use computer vision algorithms to identify and classify crops and weeds, and then decide which actions to take (e.g., spraying herbicides on weeds). The robot can also use machine learning algorithms to improve its decision-making over time.

Communication and control: The robot should be able to communicate with a central control system or with other robots in the field. This allows the robot to coordinate its actions with other robots and receive instructions from a central command center.

Power and energy management: The robot should be powered by a reliable energy source such as batteries or solar panels. The robot should also be designed to conserve energy and operate efficiently, as it may need to operate for long periods without recharging.

Overall, a successful agricultural robot system should be designed to perform specific tasks with a high degree of accuracy and efficiency, while also being adaptable to changing conditions in the field. The system should also be scalable, so that additional robots can be added as needed to cover larger areas of farmland.

ADVANTAGES:

- Precision weeding
- Cost effective
- Remote operation

3.3. SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- ✓ Arduino
- ✓ Jetson nano
- ✓ Gyroscope
- ✓ Flex sensor
- ✓ Mems
- ✓ Camera
- ✓ Bluetooth
- ✓ DC motor

1. Arduino

The Arduino Uno is an open-source microcontroller board based on the MicrochipATmega328P microcontroller and developed by Arduino.cc. The board is equipped

with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

ARDUINO:

setup( ): A function present in every Arduino sketch. Run once before the loop( ) function. Often used to set pinmode to input or output. The setup( ) function

looks like:

```
void setup( ){
    //code goes here
    }
```

loop(): A function present in every single Arduino sketch. This code happens repeatedly. The loop() is where (almost) everything happens. The one exception to this is setup() and variable declaration. ModKit uses another type of loop called "forever( )" which executes over Serial. The loop( ) function looks like:

```
void loop( ) {
    //code goes here
    }
```

input: A pin mode that intakes information.

output: A pin mode that sends information.

HIGH: Electrical signal present (5V for Uno). Also ON or True in boolean logic.

LOW: No electrical signal present (0V). Also OFF or False in boolean logic.

digitalRead: Get a HIGH or LOW reading from a pin already declared as an input.

digitalWrite: Assign a HIGH or LOW value to a pin already declared as an output.

analogRead: Get a value between or including 0 (LOW) and 1023 (HIGH). This allows you to get readings from analog sensors or interfaces that have more than two states.

analogWrite: Assign a value between or including 0 (LOW) and 255 (HIGH). This allows you to set output to a PWM value instead of just HIGH or LOW.

PWM: Stands for Pulse-Width Modulation, a method of emulating an analog signal through a digital pin. A value between or including 0 and 255. Used with **analogWrite.** These boards below use the same micro-controller, just in a different package. The Lilypad is designed for use with conductive thread instead of wire

and the Arduino Mini is simply a smaller package without the USB, Barrel Jack and Power Outs.

It depends on what you want to do with it really. There are two different purposes outlined above for the voltage divider, we will go over both.

If you wish to use the voltage divider as a sensor reading device first you need to know the maximum voltage allowed by the analog inputs you are using to read the signal. On an Arduino this is 5V. So, already we know the maximum value we need for Vout. The Vin is simply the amount of voltage already present on the circuit before it reaches the first resistor. You should be able to find the maximum voltage your sensor outputs by looking on the Datasheet; this is the maximum amount of voltage your sensor will let through given the voltage in of your circuit. Now we have exactly one variable left, the value of the second resistor. Solve for R2 and you will have all the components of your voltage divider figured out! We solve for R1's highest value because a smaller resistor will simply give us a smaller signal which will be readable by our analog inputs.

Powering an analog Reference is exactly the same as reading a sensor except you have to calculate for the Voltage Out value you want to use as the analog Reference. All of the electrical signals that the Arduino works with are either Analog or Digital. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

DIGITAL

An electronic signal transmitted as binary code that can be either the presence or absence of current, high and low voltages or short pulses at a particular frequency.

Humans perceive the world in analog, but robots, computers and circuits use Digital. A digital signal is a signal that has only two states. These states can vary depending on the signal, but simply defined the states are ON or OFF, never in between.

In the world of Arduino, Digital signals are used for everything with the exception of Analog Input. Depending on the voltage of the Arduino the ON or

HIGH of the Digital signal will be equal to the system voltage, while the OFF or LOW signal will always equal 0V.

This is a fancy way of saying that on a 5V Arduino the HIGH signals will be a little under 5V and on a 3.3V Arduino the HIGH signals will be a little under 3.3V.

To receive or send Digital signals the Arduino uses Digital pins # 0 - # 13. You may also setup your Analog In pins to act as Digital pins. To set up Analog In pins as Digital pins use the command:

- pinMode(pinNumber, value);

Where pinNumber is an Analog pin (A0 – A5) and value is either INPUT or OUTPUT. To setup Digital pins use the same command but reference a Digital pin for pinNumber instead of an Analog In pin. Digital pins default as input, so really you only need to set them to OUTPUT in pinMode. To read these pins use the command:

- digitalRead(pinNumber);

Where pinNumber is the Digital pin to which the Digital component is connected. The digitalRead command will return either a HIGH or a LOW signal. To send a Digital signal to a pin uses the command:

- digitalWrite(pinNumber, value);

Where pinNumber is the number of the pin sending the signal and value is either HIGH or LOW.

The Arduino also has the capability to output a Digital signal that acts as an Analog signal; this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, # 5, # 6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command:

- analogWrite(pinNumber, value);

Where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

THINGS TO REMEMBER ABOUT DIGITAL:

- Digital Input/Output uses the Digital pins, but Analog In pins can be used as Digital
- To receive a Digital signal use: digitalRead(pinNumber);
- To send a Digital signal use: digitalWrite(pinNumber, value);
- Digital Input and Output are always either HIGH or LOW

All of the electrical signals that the Arduino works with are either Analog or Digital. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

ANALOG

Humans perceive the world in analog. Everything we see and hear is a continuous transmission of information to our senses. The temperatures we perceive are never 100% hot or 100% cold, they are constantly changing between our ranges of acceptable temperatures. (And if they are out of our range of acceptable temperatures then what are we doing there?) This continuous stream is what defines analog data. Digital information, the complementary concept to Analog, estimates analog data using only ones and zeros.

In the world of Arduino an Analog signal is simply a signal that can be HIGH

(on), LOW (off) or anything in between these two states. This means an Analog signal has a voltage value that can be anything between 0V and 5V (unless you mess with the Analog Reference pin). Analog allows you to send output or receive input about devices that run at percentages as well as on and off. The Arduino does this by sampling the voltage signal sent to these pins and comparing it to a voltage reference signal (5V). Depending on the voltage of the Analog signal when compared to the Analog Reference signal the Arduino then assigns a numerical value to the signal somewhere between 0 (0%) and 1023 (100%). The digital system of the Arduino can then use this number in calculations and sketches.

To receive Analog Input the Arduino uses Analog pins # 0 - # 5. These pins are designed for use with components that output Analog information and can be used for Analog Input. There is no setup necessary, and to read them use the command:

- analogRead(pinNumber);

Where pinNumber is the Analog In pin to which the the Analog component is connected. The analogRead command will return a number including or between 0 and 1023.

The Arduino also has the capability to output a digital signal that acts as an Analog signal; this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, # 5, # 6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command:

- analogWrite(pinNumber, value);

Where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are signified by

a ~ sign. For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

THINGS TO REMEMBER ABOUT ANALOG:

- Analog Input uses the Analog In pins, Analog Output uses the PWM pins
- To receive an Analog signal use: analogRead(pinNumber);
- To send a PWM signal use: analogWrite(pinNumber, value);
- Analog Input values range from 0 to 1023 (1024 values because it uses 10 bits, 210)
- PWM Output values range from 0 to 255 (256 values because it uses 8 bits, 28)

All of the electrical signals that the Arduino works with are either input or output. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

OUTPUT SIGNALS

Output to the Arduino pins is always Digital, however there are two different types of Digital Output; regular Digital Output and Pulse Width Modulation Output (PWM). Output is only possible with Digital pins # 0 - # 13. The Digital pins are preset as Output pins, so unless the pin was used as an Input in the same sketch, there is no reason to use the pinMode command to set the pin as an Output. Should a situation arise where it is necessary to reset a Digital pin to Output from Input use the command:

- pinMode(pinNumber, OUTPUT);

Where pinNumber is the Digital pin number set as Output. To send a Digital Output signal use the command:

- digitalWrite(pinNumber, value);

Where pinNumber is the Digital pin that is outputting the signal and value is the signal. When outputting a Digital signal value can be either HIGH (On) or LOW (Off).

- analogWrite(pinNumber, value);

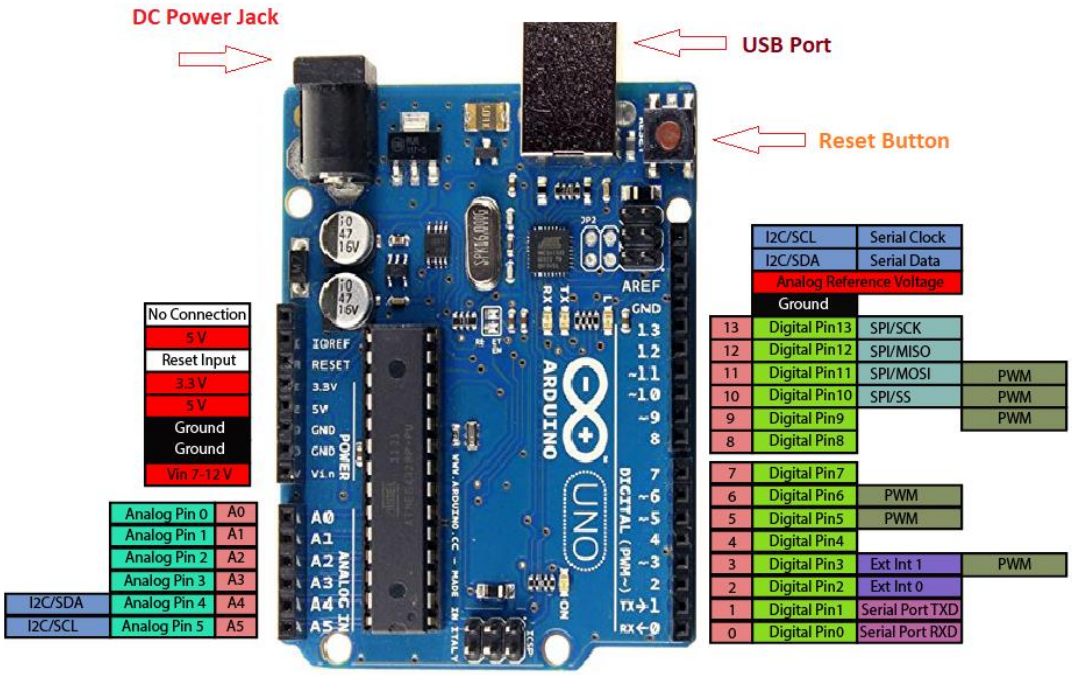
Where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

Output can be sent to many different devices, but it is up to the user to figure out which kind of Output signal is needed, hook up the hardware and then type the correct code to properly use these signals.

THINGS TO REMEMBER ABOUT OUTPUT:

- Output is always Digital
- There are two kinds of Output: regular Digital or PWM (Pulse Width Modulation)
- To send an Output signal use analogWrite(pinNumber, value); (for analog) or digitalWrite(pinNumber, value); (for digital)
- Output pin mode is set using the pinMode command: pinMode(pinNumber, OUTPUT);
- Regular Digital Output is always either HIGH or LOW
- PWM Output varies from 0 to 255

All of the electrical signals that the Arduino works with are either input or output. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.



INPUT SIGNALS

Analog Input enters your Arduino through the Analog In pins # 0 - # 5. These signals originate from analog sensors and interface devices. These analog sensors and devices use voltage levels to communicate their information instead of a simple yes (HIGH) or no (LOW). For this reason you cannot use a digital pin as an input pin for these devices. Analog Input pins are used only for receiving Analog signals. It is only possible to read the Analog Input pins so there is no command necessary in the setup( ) function to prepare these pins for input. To read the Analog Input pins use the command:

AnalogRead (pinNumber);

Where pinNumber is the Analog Input pin number. This function will return an Analog Input reading between 0 and 1023. A reading of zero corresponds to 0

Volts and a reading of 1023 corresponds to 5 Volts. These voltage values are emitted by the analog sensors and interfaces. If you have an Analog Input that could exceed Vcc + .5V you may change the voltage that 1023 corresponds to by using the Aref pin. This pin sets the maximum voltage parameter your Analog Input pins can read. The Aref pin's preset value is 5V.

Digital Input can enter your Arduino through any of the Digital Pins # 0 - # 13. Digital Input signals are either HIGH (On, 5V) or LOW (Off, 0V). Because the Digital pins can be used either as input or output you will need to prepare the Arduino to use these pins as inputs in your
setup( )function. To do this types the command:

- pinMode(pinNumber, INPUT);

inside the curly brackets of the setup( ) function where pinNumber is the Digital pin number you wish to declare as an input. You can change the pinMode in the loop( )function if you need to switch a pin back and forth between input and output, but it is usually set in the setup( )function and left untouched in the loop( )function. To read the Digital pins set as inputs use the command:

- digitalRead(pinNumber);

Where pinNumber is the Digital Input pin number.

Input can come from many different devices, but each device's signal will be either Analog or Digital, it is up to the user to figure out which kind of input is needed, hook up the hardware and then type the correct code to properly use these signals.

X1:

- ✓ DE-9 serial connector

Used to connect computer (or other devices) using RS-232 standard. Needs a serial cable, with at least 4 pins connected: 2, 3, 4 and 5.Works only when JP0 is set to 2-3 position.

DC1:

- ✓ 2.1 mm. power jack

Used to connect external power source. Centre positive. Voltage Regulator Works with regulated +7 to +20 volts DC (9v. to 12v. is recommended). It is possible to alternatively connect external power using 9v. pin or 5v. pin. (see POWER PINOUT)

ICSP:

2x3 pin header Used to program Atmega with bootloader. The number 1 on both sides of the board indicates cable pin1 position. Used to upload sketches on Atmega ICs without bootloader (available only in Arduino IDE versions 0011 and 0012).

JP0

3 pins jumper When in position 2-3, this jumper enables serial connection (through X1 connector) to/from computer/devices. Use this as default position. When in position 1-2, it disables serial communication, and enables external pull-down resistors on pin0 (RX) and pin1 (TX). Use this only to prevent noise on RX (that seems incoming data to Atmega), that sometimes makes sketch not starting. When removing this jumper, serial communication is disabled, and pin0 and pin1 work as a normal (floating) digital pin. Useful when more digital pins are needed, but only when serial communication is not necessary. External pull-down/pull-up resistor is required.

JP4

2 pins jumper When in position 1-2, this jumper enables auto reset feature, useful when uploading a sketch to Arduino, resetting Atmega automatically. It makes unnecessary to press reset button (S1) when uploading sketches. Be sure that computer COM Port speed is set to 19200bps otherwise auto reset will not work

properly. If removed, disables auto reset feature. Very useful to prevent undesired Atmega reset when using sketches that needs serial communication. Auto reset works with DTR pulse on serial pin4. Sometimes Arduino senses a DTR pulse when connecting X1 (serial connector) and some softwares sends a DTR pulse when it starts or when it closes, that makes Atmega reset when not desired.

S1

Tactile button This button resets Atmega, to restart uploaded sketch or to prepare Arduino to receive a sketch through serial connector (when auto reset is not active).

LEDS

Indicative leds POWER led Turns on when Arduino is powered through DC1, +9v. pin or +5v. pin. RX led Blinks when receiving data from computer/device through serial connection. TX led Blinks when sending data to computer/device through serial connection. L led This led is connected to digital pin13 with a current limiter resistor (that doesn't affect pin13). Useful to test sketches. It is normal to blink when bootloading too.

POWER PINOUT

6 pin header

RST pin

Makes Atmega reset when connected to GND. Useful for Shield Boards, or to connect external reset.

NC pin

This pin is not connected in Arduino S3v3. Arduino Diecimila has a 3.3 volts pin in the same position.

+9v pin

When Arduino DC1 is powered (with battery or DC adaptor), this pin is used as

Vout, with the same voltage supplied on DC1 (see DC1), minus 0,7 volts. The total supplied current depends on external power source capacity When Arduino DC1 is not powered, +9v. pin can be used as Vin, connecting it to a external regulated power source (+7 to +20 volts) and connecting 0v. pin to external power source GND. In this case,

+5v pin

It can be used as Vout, supplying +5 volts. +5v.pin When Arduino DC1 is powered (with battery or DC adaptor), +5v. pin supplies +5 volts as a Vout pin. The total supplied current depends on Voltage Regulator (7805 supplies up to 1A). This applies only to +5v. pin: Atmega in/out pins only supplies max. 40mA on each pin. When Arduino DC1 is not powered, this pin can be used as Vin, connecting it to a regulated +5v. and connecting 0v. pin to power source GND. In this case, +9v.pin is inactive. 0v. pin (GND) Two 0v. pins between +5v. and +9v. / One

0v. pin

Beside AREF pin.When Arduino DC1 is powered, 0v.pin supplies 0 volts reference (GND) for +5v. pin and +9v. pin. When DC1 is not powered, and Arduino is powered through +5v.pin or +9v. pin, 0v. pin must be used as GND reference, connecting it to the external power source GND.

GND pin

see 0v. pin (GND).

AREF pin

The AREF can be set to AVcc (default), internal 2.56 volts (Atmega8), internal 1.1 volts (Atmega168), or external AREF. In case of AVcc or internal AREF, AREF pin can be used to attach na external capacitor to decouple the signal, for better noise performance. In case of external AREF, AREF pin is used to attach the external reference voltage. Remember that it is necessary to change de fuses (wiring.c file), and re-upload sketch, before connecting external voltage to AREF

SOFTWARE TIPS

When bootloadingna Atmega8 chip with Arduino 0010, there is a command (-i800) that makes bootloader delay 10 minutes. So, if you need to use bootloader, use command line instead of IDE, removing "–i800" command and adding "–F" command, or use Arduino 0007 IDE. To upload sketches Arduino 0010 works fine.

ARDUINO S3v3 NEW FEATURES

Full compatible with Shield Boards (Version 2 is the only Arduino Board not compatible with Shield Boards because of ICSP header wrong position, and tall components);

- AVcc LP filter to reduce noise level on ADC;
- auto reset feature;
- auto reset enable/disable jumper, to avoid not desired reseting;
- arduinoDiecimila compatible reset pin;
- pin13 onboard led, with current limiter resistor;
- TX and RX onboard leds;
- power led with appropriate current limiter resistor (less 20mA of comsumption);
- jumper to disable serial communication and to enable RX external pull down resistor, to avoid "RX floating error". This feature allows to use digital pin0 and pin1 as a normal pin, when serial communication is not needed;
- all similar components (diodes, transistors, leds, capacitors) has the same board orientation (to makes easier to mount with less mistakes);
- no wires between pads, more space between wires, larger wires, larger pads (better for etching, soldering and drilling, with no short circuits, soldering bridges or open wires in corrosion);

- only 3 wire bridges;

- electrolytic capacitor (in serial to TTL circuit) changed to bipolar type (to avoid inverted voltage problem when serial cable is not connected);
- All jumpers are right angle type, to allow Shield Boards use.

2. JETSON NANO:

The Jetson Nano is a compact and powerful computer designed to enable the development of artificial intelligence (AI) and machine learning (ML) applications at the edge. It is part of the Jetson family of products from NVIDIA, a leader in AI computing technology. The Jetson Nano is an affordable, low-power development kit that can be used for prototyping, testing, and deploying AI applications.

The Jetson Nano is built around an NVIDIA Tegra X1 SoC, which includes a Quad-core ARM Cortex-A57 CPU and a 128-core NVIDIA Maxwell GPU. It also has 4GB of LPDDR4 memory and supports several connectivity options, including Gigabit Ethernet, USB 3.0, and MIPI-CSI camera connector. The Jetson Nano can run various AI frameworks, including TensorFlow, PyTorch, and Caffe, and can

be programmed using Python, C++, and CUDA.

The Jetson Nano is designed to be used in a wide range of applications, including robotics, drones, IoT devices, and smart cameras. Its low power consumption, compact form factor, and

high processing power make it an ideal platform for running AI workloads at the edge. Some of the applications that can be built using the Jetson Nano include object detection, image classification, speech recognition, and autonomous navigation.

One of the key advantages of the Jetson Nano is its ease of use. The development kit comes with pre-installed software, including the JetPack SDK, which includes several libraries and tools for developing AI applications. The SDK also includes a comprehensive set of documentation and tutorials that can help developers get started quickly. Additionally, the Jetson Nano is compatible with various sensors and peripherals, which makes it easy to interface with other hardware components.

Another advantage of the Jetson Nano is its affordability. The development kit is priced at a fraction of the cost of other AI computing platforms, making it accessible to a broader range of developers and hobbyists. The low cost also makes it an attractive option for organizations looking to develop and deploy AI applications at scale.

In conclusion, the Jetson Nano is a powerful and versatile development kit that provides an accessible platform for building AI-powered applications at the edge. Its low cost, ease of use, and high processing power make it an ideal choice for developers and organizations looking to explore the potential of AI and machine learning.

3. GYROSCOPE:

A gyroscope is a device that measures or maintains orientation and angular

velocity. It consists of a spinning wheel or rotor that is mounted on a set of gimbals so that it can rotate freely in any direction. When the rotor is spun up, it resists any change in its orientation due to the principle of conservation of angular momentum. Gyroscopes are used in a wide variety of applications, ranging from navigation and stabilization systems in aircraft and spacecraft to motion sensors in smartphones and gaming controllers. They are also used in inertial navigation systems, which can provide accurate positioning and velocity information without relying on external sources such as GPS.

Gyroscopes work on the principle of gyroscopic precession, which is the tendency of a spinning object to experience a torque when a force is applied perpendicular to the plane of rotation. This torque causes the rotor to rotate around its axis, which in turn causes the gimbals to rotate around their axes. The rate of precession is proportional to the magnitude of the applied force and the angular velocity of the rotor. Gyroscopes can be classified into two main types: mechanical gyroscopes and laser gyroscopes. Mechanical gyroscopes use a spinning mass to measure or maintain orientation, while laser gyroscopes use the interference of laser beams to detect angular velocity.

Mechanical gyroscopes can be further classified into three types: spinning mass

gyroscopes, vibrating gyroscopes, and fiber optic gyroscopes. Spinning mass gyroscopes use a spinning mass to maintain orientation, while vibrating gyroscopes use the Coriolis effect to detect angular velocity. Fiber optic gyroscopes use the interference of light waves to detect angular velocity.

Laser gyroscopes can also be classified into two types: ring laser gyroscopes and fiber optic gyroscopes. Ring laser gyroscopes use the interference of laser beams to measure angular velocity, while fiber optic gyroscopes use the interference of light waves in a coiled fiber optic cable to detect angular velocity.

In addition to navigation and stabilization systems, gyroscopes are used in a wide variety of other applications, including robotics, gaming, and virtual reality. They are also used in medical devices such as endoscopes and surgical instruments.

Overall, gyroscopes are a fascinating and important technology that have revolutionized many fields of study and application. From their early beginnings in mechanical devices to their current use in laser-based systems, gyroscopes have become a critical component in many modern technologies and will likely continue to play an important role in the future.

4. FLEX SENSOR

A flex sensor is a kind of sensor which is used to measure the amount of defection otherwise bending. The designing of this sensor can be done by using materials like plastic and carbon. The carbon surface is arranged on a plastic strip as this strip is turned aside then the sensor’s resistance will be changed. Thus, it is also named a bend sensor. As its varying resistance can be directly proportional to the quantity of turn thus it can also be employed like a goniometer.

We know that there are different types of sensors available in the market where each sensor can be used based on the application. Likewise, a bend sensor or flex sensor is one kind of sensor used to measure the quantity of bending otherwise deflection. Generally, this sensor is fixed to the exterior, and the resistance of this sensor can be changed by twisting the exterior. These sensors are applicable in the Nintendo power glove, robot whisker sensors, door sensors, otherwise the main component in making alert stuffed animal toys.

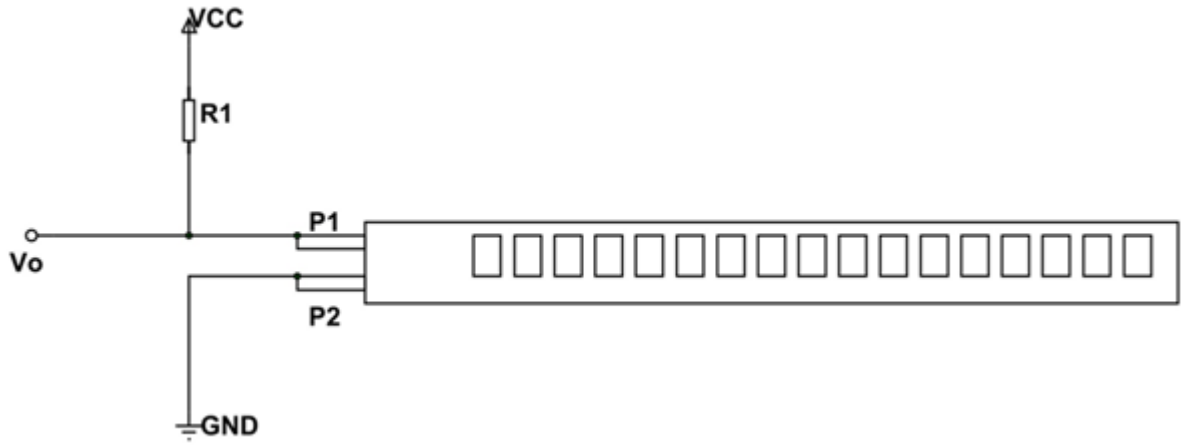
Types of Flex Sensor

These sensors are classified into two types based on its size namely 2.2-inch flex sensor & 4.5-inch flex sensor. The size, as well as the resistance of these sensors, is dissimilar except the working principle.

Therefore, the suitable size can be preferred based on the necessity. Here this article discusses an overview of 2.2-inch flex-sensor. This type of sensor is used in various applications like computer interface, rehabilitation, security system, music interface, intensity control, and wherever the consumer needs to modify the resistance throughout bending.

Pin Configuration

The pin configuration of the flex sensor is shown below. It is a two-terminal device, and the terminals are like p1 & p2. This sensor doesn't contain any polarized terminal such as diode otherwise capacitor, which means there is no positive & negative terminal. The required voltage of this sensor to activate the sensor ranges from 3.3V -5V DC which can be gained from any type of interfacing.



- Pin P1: This pin is generally connected to the +ve terminal of the power source.
- Pin P2: This pin is generally connected to GND pin of the power source.

Where to Use?

The flex-sensor can be used in the following two cases.

Case1: Where you want to check whether the surface of a device or thing is leveled or not. Say you want a device to check whether a window or door is open or not. At that time a Flex sensor could be used. The sensor could be fixed at door edge and when the door opens the Flex sensor gets flexed. With the sensor being flexed its parameters changes which could be designed to provide an alert.

Case2: Where you want to measure the FLEX or BENT or ANGLE change of any instrument or device. The FLEX SENSOR internal resistance changes almost linearly with its flex angle. So by sticking the sensor to the instrument we can have the flex angle in electrical parameter of resistance.

How to Use FLEX SENSOR

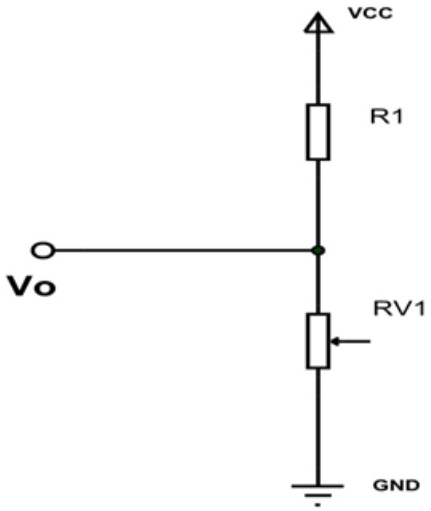
As mentioned earlier, **FLEX SENSOR** is basically a **VARIABLE RESISTOR** whose terminal resistance increases when the sensor is bent. So this sensor resistance increases depends on surface linearity. So it is usually used to sense the changes in linearity.

This sensor works on the bending strip principle which means whenever the strip is twisted then its resistance will be changed. This can be measured with the help of any controller.

This sensor works similar to a variable resistance because when it twists then the resistance will be changed. The resistance change can depend on the linearity of the surface because the resistance will be dissimilar when it is level.

When the sensor is twisted 450 then the resistance would be dissimilar. Similarly, when this senor is twisted to 900 then the resistance would be dissimilar. These three are the flex sensor's bending conditions.

According to these three cases, the resistance will be normal in the first case, the resistance will be double as contrasted with the first case, and the resistance will be four-time when compared with the first case. So the resistance will be increased when the angle is increased.



Specifications & Features

The specifications and features of this sensor include the following.

- Operating voltage of this sensor ranges from 0V to 5V
- It can function on low-voltages.
- Power rating is 1 Watt for peak & 0.5Watt for continuous.
- Operating temperature ranges from -45ºC to +80ºC
- Flat resistance is 25K Ω
- The tolerance of resistance will be ±30%
- The range of bend resistance will range from45K -125K Ohms

Applications

The applications of the flex-sensor include the following.

- Medical Instruments
- Peripherals of Computer
- Robotics
- Physical Therapy
- Virtual Motion (Gaming)
- Musical Instrument.

6. MEMS SENSOR

Micro-electro-mechanical Systems (MEMS) Technology is one of the most advanced technologies that have been applied in the making of most of the modern devices like video projectors, bi-analysis chips and also car crash airbag sensors. This concept was first explained by Professor R. Howe in the year 1989. Since then many prototypes have been released and revised and has thus become an integral part of the latest mechanical products available in the market today.

During its early stage, the MEMS chip had two parts. One part included the main structure of the chip and the other part included everything needed for signal conditioning. This method was not successful as the total space taken by the device was larger, and thus the different parts of a single chip needed multi-assembling procedures. The output obtained from such a device had less accuracy and the mounting of such a device was difficult.

As the technology became more advanced the idea of integrating multi-chips was applied on to produce a single chip MEMS with high performance and accuracy.

The main idea behind this technology is to use some of the basic mechanical devices like cantilevers and membranes to have the same qualities of electronic circuits. To obtain such a concept, micro-fabrication process must be carried out. Though an electronic process is carried out, an MEMS device cannot be called as an electronic circuit. MEMS duplicate a mechanical part and have holes, cantilevers, membranes, channels, and so on. But an electronic circuit has a firm and compact structure. To make MEMS from silicon process, the manufacturer must have a deep knowledge in electronics, mechanical and also about the materials used for the process.

MEMS Accelerometer

An accelerometer is an electromechanical device that is used to measure acceleration and the force producing it. Many types of accelerometers are available in the market today. They can be divided according to the force (static or dynamic) that is to be measured. Even today, one of the most commonly used one is the piezoelectric accelerometer. But, since they are bulky and cannot be used for all operations, a smaller and highly functional device like the MEMS accelerometer was developed. Though the first of its kind was developed 25 years ago, it was not accepted until lately, when there was need for large volume industrial applications. Due to its small size and robust sensing feature, they are further developed to obtain multi-axis sensing.

Working

One of the most commonly used MEMS accelerometer is the capacitive type. The capacitive MEMS accelerometer is famous for its high sensitivity and its accuracy at high temperatures. The device does not change values depending on the base materials used and depends only on the capacitive value that occurs due to the change in distance between the plates.

If two plates are kept parallel to each other and are separated by a distance‘d’, and if ‘E’ is the permitivity of the separating material, then capacitance produced can be written as

$$\mathbf{C_0 = E_0.E\ A/d = E_A/d}$$

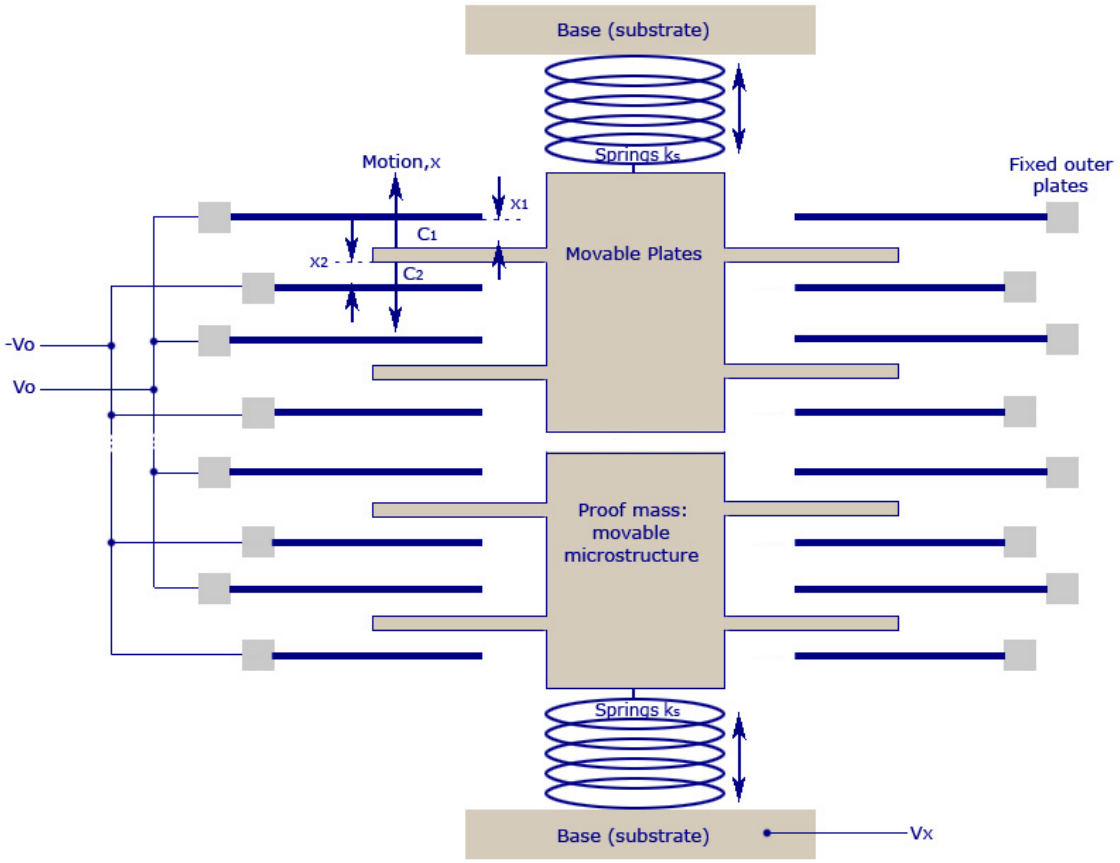
$$\mathbf{E_A = E_0EA}$$

A – Area of the electrodes

A change in the values of E, A or d will help in finding the change in capacitance and thus helps in the working of the MEMS transducer. Accelerometer values mainly depend on the change of values of d or A.

A typical MEMS accelerometer is shown in the figure below. It can also be called a simple one-axis accelerometer. If more sets of capacitors are kept in 90

degrees to each other you can design 2 or 3-axis accelerometer. A simple MEMS transducer mainly consists of a movable microstructure or a proof mass that is connected to a mechanical suspension system and thus on to a reference frame.



The movable plates and the fixed outer plates act as the capacitor plates. When acceleration is applied, the proof mass moves accordingly. This produces a capacitance between the movable and the fixed outer plates.

When acceleration is applied, the distance between the two plates displace as X1 and X2, and they turn out to be a function of the capacitance produced.

From the image above it is clear that all sensors have multiple capacitor sets. All upper capacitors are wired parallel to produce an overall capacitance C1 and the lower ones produce an overall capacitance of C2.

If V_x is the output voltage of the proof mass, and V_0 is the output voltage

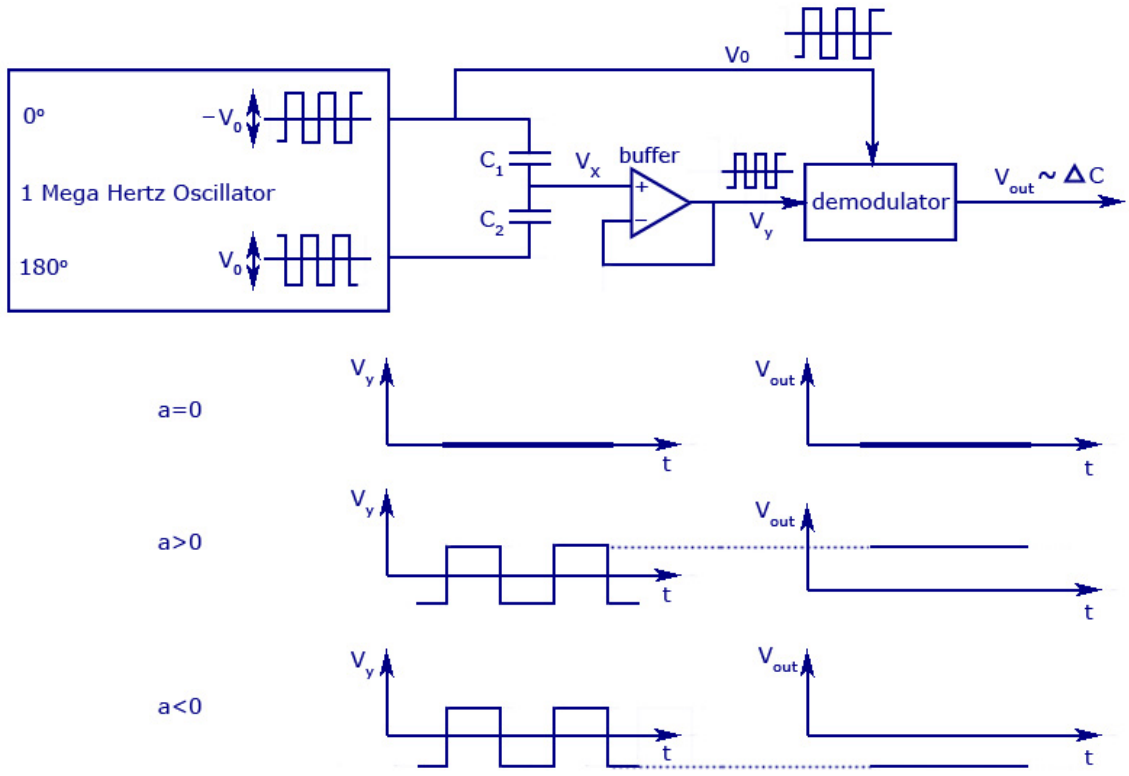
produced between the plates, then

$$(V_x + V_0)\, C1 + (V_x - V_0)\, C2 = 0$$

We can also write

$$V_x = V_0\,[(C2-C1)/(C2+C1)] = (x/d)\, V_0$$

The figure below shows the circuit that is used to calculate the acceleration, through change in distance between capacitor plates. The output obtained for different values of acceleration is also shown graphically.



Acceleration Calculation - Capacitor Type MEMS Accelerometer

When no acceleration is given (a=0), the output voltage will also be zero.

When acceleration is given, such as (a>0), the value of value of V_x changes in proportion to the value of V_0.

When a deceleration is given, such as (a<0), the signals V_x and V_y become negative.

He demodulator produces an output equal to the sign of the acceleration, as it multiplies both the values of V_y and V_0 to produce V_{OUT}, which has the correct acceleration sign and correct amplitude.

The length of the distance, d and the proof mass weigh is surprisingly very small. The proof mass weighs no more than 0.1 microgram and the output capacitance is approximately 20 aF and the plate distance is no more than 1.3 micrometers.

We must select the device in reference to its noise characteristics. If the acceleration value at low gravity condition is to be found out, the noise characteristics could easily affect its accuracy. An MEMS accelerometer is said to have three noise producing parameters – from the signal conditioning circuit, from the vibrations produced by the springs, and from the output measuring system.

MEMS Accelerometers – Applications

- MEMS sensors are being used in latest mobile phones and gaming joysticks as step counters, user interface control, and also for switching between different modes.
- Used in mobile cameras as a tilt sensor so as to tag the orientation of photos taken.
- To provide stability of images in camcorders and also to rotate the image to and fro when you turn the mobile.
- A 3D accelerometer is used in Nokia 5500 so as to provide easier tap and change feature by which you can change mp3's by tapping on the phone when it is lying inside the pocket.
- Used to protect hard disk drives in laptops from getting damaged when the PC falls to the ground. The device senses the free fall and automatically switches off the hard disk.
- Used in car crash airbag sensors, where it senses the sudden negative acceleration and determines the correct time to open the airbag.

- Used in real-time applications like military monitoring, missile launching, projectiles.

Advantages

1. MEMS device are very small and can be applicable for many mechanical purposes where large measurements are needed.
2. The small size of the device has also helped in reducing its cost.
3. If two or three different devices are needed to deploy a particular process, all of them can be easily integrated in an MEMS chip with the help of microelectronics. Thus, data reception, filtering, storing, transfer, interfacing, and all other processes can be carried out with a single chip.

Applications

1. The device is highly applicable as an accelerometer, and thus can be deployed as airbag sensors or in digital cameras in order to stabilize the image.
2. Can be used as a pressure sensor to calculate the pressure difference in blood, manifold pressure (MAP), and also tire pressure.
3. It is commonly used in a gyroscope, DNA chips and inkjet printer nozzle.
4. Optical MEMS is used for making projectors, optical fiber switch and so on.

7. CAMERA

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video. For example, Apple's iSight camera, which is built into Apple laptops, iMacs and

a number of iPhones, can be used for video chat sessions, using the iChat instant messaging program (now called Messages). Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires a lot of bandwidth, such streams usually use compressed formats. The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions

Optics

Various lenses are available, the most common in consumer-grade webcams being a plastic lens that can be manually moved in and out to focus the camera. Fixed-focus lenses, which have no provision for adjustment, are also available. As a camera system's depth of field is greater for small image formats and is greater for lenses with a large f-number (small aperture), the systems used in webcams have a sufficiently large depth of field that the use of a fixed-focus lens does not impact image sharpness to a great extent.

Most models use simple, focal-free optics (fixed focus, factory-set for the usual distance from the monitor to which it is fastened to the user) or manual focus.

Compression

Digital video streams are represented by huge amounts of data, burdening its transmission (from the image sensor, where the data is continuously created) and storage alike.

Most if not all cheap webcams come with built-it ASIC to do video compression in real-time.

Support electronics read the image from the sensor and transmit it to the host computer. The camera pictured to the right, for example, uses a Sonix SN9C101 to transmit its image over USB. Typically, each frame is transmitted uncompressed in RGB or YUV or compressed as JPEG. Some cameras, such as mobile-phone cameras, use a CMOS sensor with supporting electronics "on die", i.e. the sensor and the support electronics are built on a single silicon chip to save space and manufacturing costs. Most webcams feature built-in microphones to make video calling and videoconferencing more convenient

Interface

Typical interfaces used by articles marketed as a "webcam" are USB, Ethernet and IEEE (denominated as IP camera). Further interfaces such as e.g. Composite video or S-Video are also available

The USB video device class (UVC) specification allows inter-connectivity of webcams to computers without the need for proprietary device drivers.

Software

Various proprietary as well as free and open-source software is available to handle the UVC stream. One could use Guvcview or GStreamer and GStreamer-based software to handle the UVC stream.

Characteristic:

Webcams are known for their low manufacturing cost and their high flexibility, making them the lowest-cost form of video telephony. As webcams

evolved simultaneously with display technologies, USB interface speeds and broadband internet speeds, the resolution went up from gradually 320×240, to 640×480, and some even offering 1280×720 (aka 720p) or 1920×1080 (aka 1080p) resolution.

Despite the low cost, the resolution offered as of 2019 is impressive, with now the low-end webcams offering resolutions of 720p, mid-range webcams offering 1080p resolution, and high-end webcams offering 4K resolution at 60 fps.

Webcams have become a source of security and privacy issues, as some built-in webcams can be remotely activated by spyware. To address this concern, many webcams come with a physical lens cover.

Uses:

The most popular use of webcams is the establishment of video links, permitting computers to act as videophones or videoconference stations. Other popular uses include security surveillance, computer vision, video broadcasting, and for recording social videos.

The video streams provided by webcams can be used for a number of purposes

Health care

Most modern webcams are capable of capturing arterial pulse rate by the use of a simple algorithmic trick. Researchers claim that this method is accurate to ±5 bpm.

Video monitoring

Webcams may be installed at places such as childcare centres, offices, shops and private areas to monitor security and general activity.

Commerce

Webcams have been used for augmented reality experiences online. One such function has the webcam act as a "magic mirror" to allow an online shopper to view a virtual item on themselves. The Webcam Social Shopper is one example of software that utilizes the webcam in this manner.

Video calling and video conferencing

Further information: Videophone, Videoconferencing, and Video telephony

Webcam can be added to instant messaging, text chat services such as AOL Instant Messenger, and VoIP services such as Skype, one-to-one live video communication over the Internet has now reached millions of mainstream PC users worldwide. Improved video quality has helped webcams encroach on traditional video conferencing systems. New features such as automatic lighting controls, real-time enhancements (retouching, wrinkle smoothing and vertical stretch), automatic face tracking and autofocus, assist users by providing substantial ease-of-use, further increasing the popularity of webcams.

Webcam features and performance can vary by program, computer operating system, and also by the computer's processor capabilities. Video calling support has also been added to several popular instant messaging programs.

Video security

Webcams can be used as security cameras. Software is available to allow PC-connected cameras to watch for movement and sound, recording both when they are detected. These recordings can then be saved to the computer, e-mailed, or uploaded to the Internet. In one well-publicised case, a computer e-mailed images of the burglar during the theft of the computer, enabling the owner to give police a clear picture of the burglar's face even after the computer had been stolen.

Unauthorized access of webcams can present significant privacy issues (see "Privacy" section below).

Video clips and stills

Webcams can be used to take video clips and still pictures. Various software tools in wide use can be employed for this, such as PicMaster (for use with Windows operating systems), Photo Booth (Mac), or Cheese (with Unix systems). For a more complete list see Comparison of webcam software.

Input control devices

Special software can use the video stream from a webcam to assist or enhance a user's control of applications and games. Video features, including faces, shapes, models and colors can be observed and tracked to produce a corresponding form of control. For example, the position of a single light source can be tracked and used to emulate a mouse pointer, a head-mounted light would enable hands-free computing and would greatly improve computer accessibility. This can be applied to games, providing additional control, improved interactivity and immersiveness.

FreeTrack is a free webcam motion-tracking application for Microsoft Windows that can track a special head-mounted model in up to six degrees of freedom and output data to mouse, keyboard, joystick and FreeTrack-supported games. By removing the IR filter of the webcam, IR LEDs can be used, which has the advantage of being invisible to the naked eye, removing a distraction from the user. TrackIR is a commercial version of this technology.

The EyeToy for the PlayStation 2, PlayStation Eye for the PlayStation 3, and the Xbox Live Vision camera and Kinect motion sensor for the Xbox 360 and are color digital cameras that have been used as control input devices by some games.

Small webcam-based PC games are available as either standalone executables or inside web browser windows using Adobe Flash.

8.HC-05 BLUETOOTH MODULE

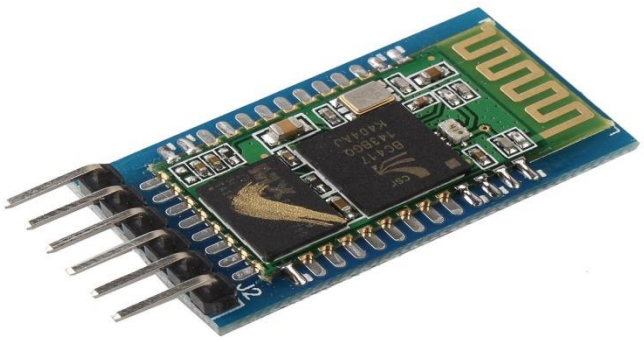
HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

HC-05 module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

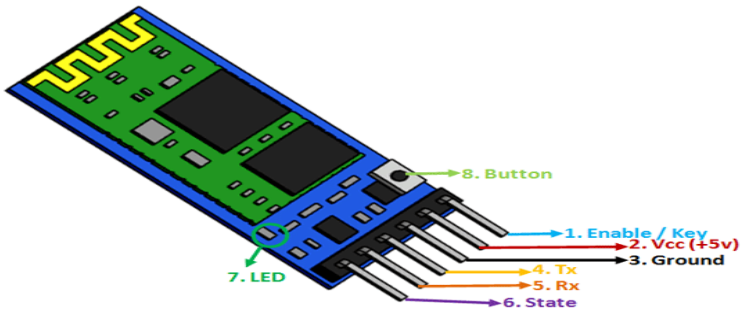
Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband.

Through which one can build wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over air.

It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).



Pin Description



Pin No	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.

7	LED	Indicates the status of Module • Blink once in 2 sec: Module has entered Command Mode • Repeated Blinking: Waiting for connection in Data Mode • Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

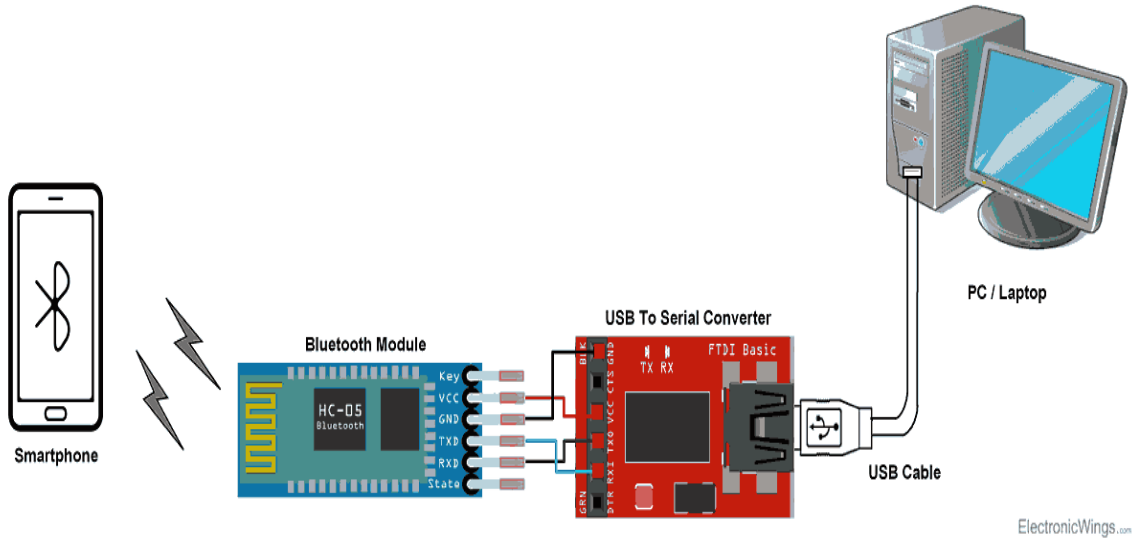
HC-05 module Information

- HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.
- This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.
- As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

Bluetooth communication between Devices

- E.g. Send data from Smartphone terminal to HC-05 Bluetooth module and see this data on PC serial terminal and vice versa.

- To communicate smartphone with HC-05 Bluetooth module, smartphone requires Bluetooth terminal application for transmitting and receiving data. You can find Bluetooth terminal applications for android and windows in respective app. store.



Bluetooth Module Serial Interface

So, when we want to communicate through smartphone with HC-05 Bluetooth module, connect this HC-05 module to the PC via serial to USB converter.

Before establishing communication between two Bluetooth devices, 1st we need to pair HC-05 module to smartphone for communication.

HC-05 Default Settings

- Default Bluetooth Name: “HC-05”
- Default Password: 1234 or 0000
- Default Communication: Slave
- Default Mode: Data Mode
- Data Mode Baud Rate: 9600, 8, N, 1
- Command Mode Baud Rate: 38400, 8, N, 1
- Default firmware: LINVOR

Pair HC-05 and smartphone:

1. Search for new Bluetooth device from your phone. You will find Bluetooth device with “HC-05” name.
2. Click on connect/pair device option; default pin for HC-05 is 1234 or 0000.

After pairing two Bluetooth devices, open terminal software (e.g. Teraterm, Realterm etc.) in PC, and select the port where we have connected USB to serial module. Also select default baud rate of 9600 bps.

In smart phone, open Bluetooth terminal application and connect to paired device HC-05.

It is simple to communicate, we just have to type in the Bluetooth terminal application of smartphone. Characters will get sent wirelessly to Bluetooth module HC-05. HC-05 will automatically transmit it serially to the PC, which will appear on terminal. Same way we can send data from PC to smartphone.

Command Mode

- When we want to change settings of HC-05 Bluetooth module like change password for connection, baud rate, Bluetooth device’s name etc.
- To do this, HC-05 has AT commands.
- To use HC-05 Bluetooth module in AT command mode, connect “Key” pin to High (VCC).
- Default Baud rate of HC-05 in command mode is 38400bps.

- Following are some AT command generally used to change setting of Bluetooth module.

- To send these commands, we have to connect HC-05 Bluetooth module to the PC via serial to USB converter and transmit these command through serial terminal of PC.

Command	Description	Response
AT	Checking communication	OK
AT+PSWD=XXXX	Set Password e.g. AT+PSWD=4567	OK
AT+NAME=XXXX	Set Bluetooth Device Name e.g. AT+NAME=MyHC-05	OK
AT+UART=Baud rate, stop bit, parity bit	Change Baud rate e.g. AT+UART=9600,1,0	OK
AT+VERSION?	Respond version no. of Bluetooth module	+Version: XX OK e.g. +Version: 2.0 20130107 OK

AT+ORGL	Send detail of setting done by manufacturer	Parameters: device type, module mode, serial parameter, passkey,etc.

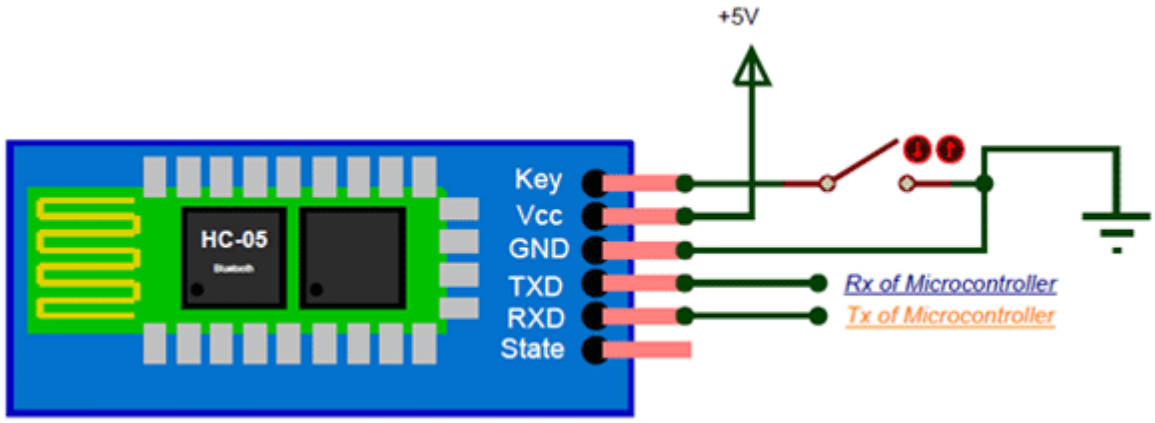
Where to use HC-05 Bluetooth module

The **HC-05** is a very cool module which can add two-way (full-duplex) wireless functionality to your projects. You can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to interface with any microcontroller that supports USART. We can also configure the default values of the module by using the command mode. So if you looking for a Wireless module that could transfer data from your computer or mobile phone to microcontroller or vice versa then this module might be the right choice for you. However do not expect this module to transfer multimedia like photos or songs; you might have to look into the CSR8645 module for that.

How to Use the HC-05 Bluetooth module

The **HC-05** has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description.

It is very easy to pair the HC-05 module with microcontrollers because it operates using the Serial Port Protocol (SPP). Simply power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as shown in the figure below



During power up the key pin can be grounded to enter into Command mode, if left free it will by default enter into the data mode. As soon as the module is powered you should be able to discover the Bluetooth device as "HC-05" then connect with it using the default password 1234 and start communicating with it. The name password and other default parameters can be changed by entering into the

Specifications

- Serial Bluetooth module for Arduino and other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)

- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

Applications

1. Wireless communication between two microcontrollers

2. Communicate with Laptop, Desktops and mobile phones

3. Data Logging application

4. Consumer applications

5. Wireless Robots

6. Home Automation

CHAPTER 4

SYSTEM DEVELOPMENT

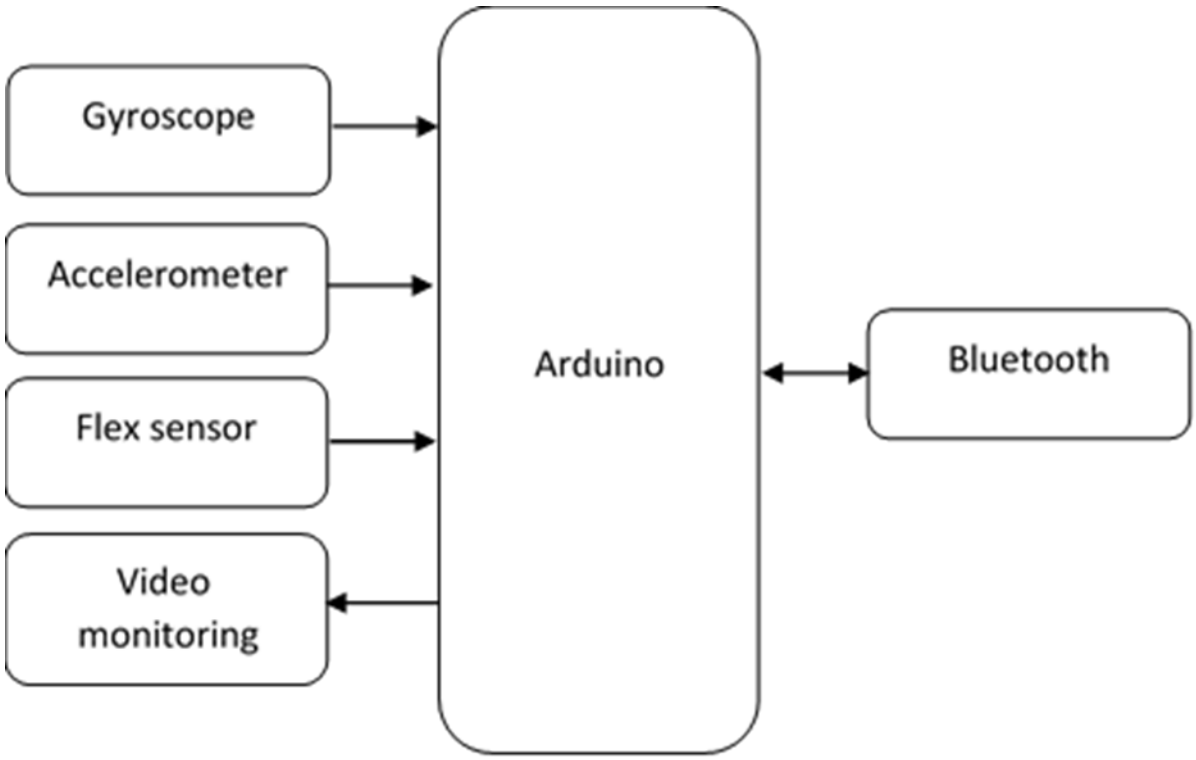
CHAPTER 4

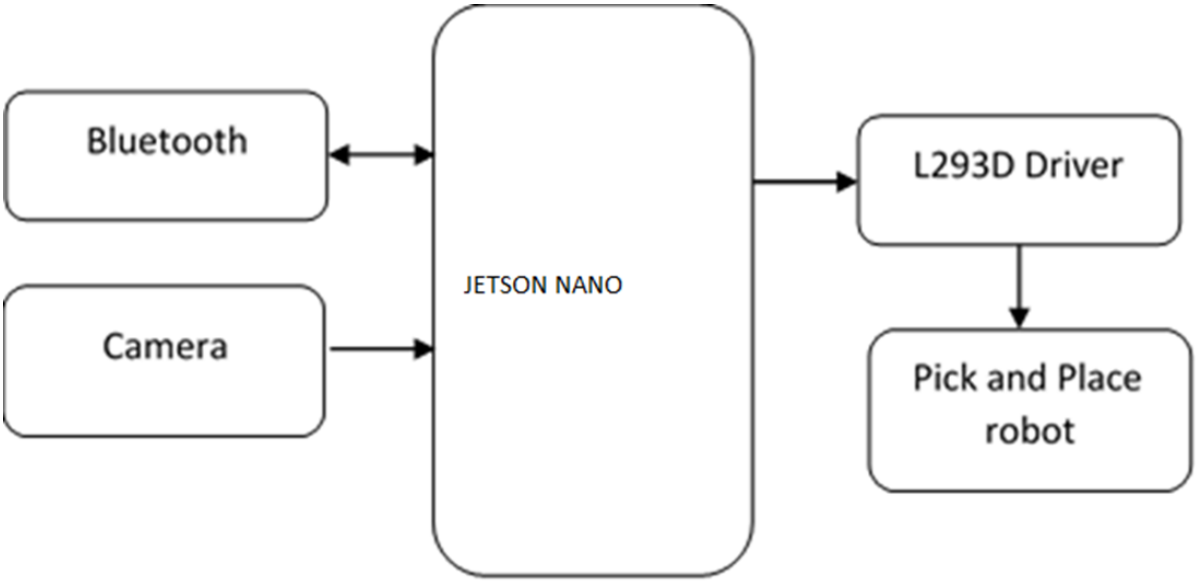
SYSTEM DEVELOPMENT

4.1 SYSTEM ARCHITECTURE

BLOCK DIAGRAM

Fig. 4.1





Block diagram explanation:

The system architecture of the proposed glove, harvester and interface. The glove is the Human Control Interface (HCI). The user wears the glove which is embedded with accelerometer, gyrometer and flex sensors. The accelerometer gives the acceleration and tilt and the gyrometer provides the angular velocity and orientation. The harvester is the rover over which the arm is fixed. The sensors are embedded in the hand glove and are interfaced to the microcontroller unit (MCU). Three accelerometers to control the 3 links of the 4 DoF arm and two flex sensors to control the cutter are embedded in the glove. The user HCI is the transmitter and the robotic harvester is the receiver. The transmitter block contains all the sensors, Bluetooth and an MCU, all integrated into a wearable device. The wearable device transmits real time joint angles of the users arm to the robotic arm.

4.2 DATA FLOW DIAGRAM

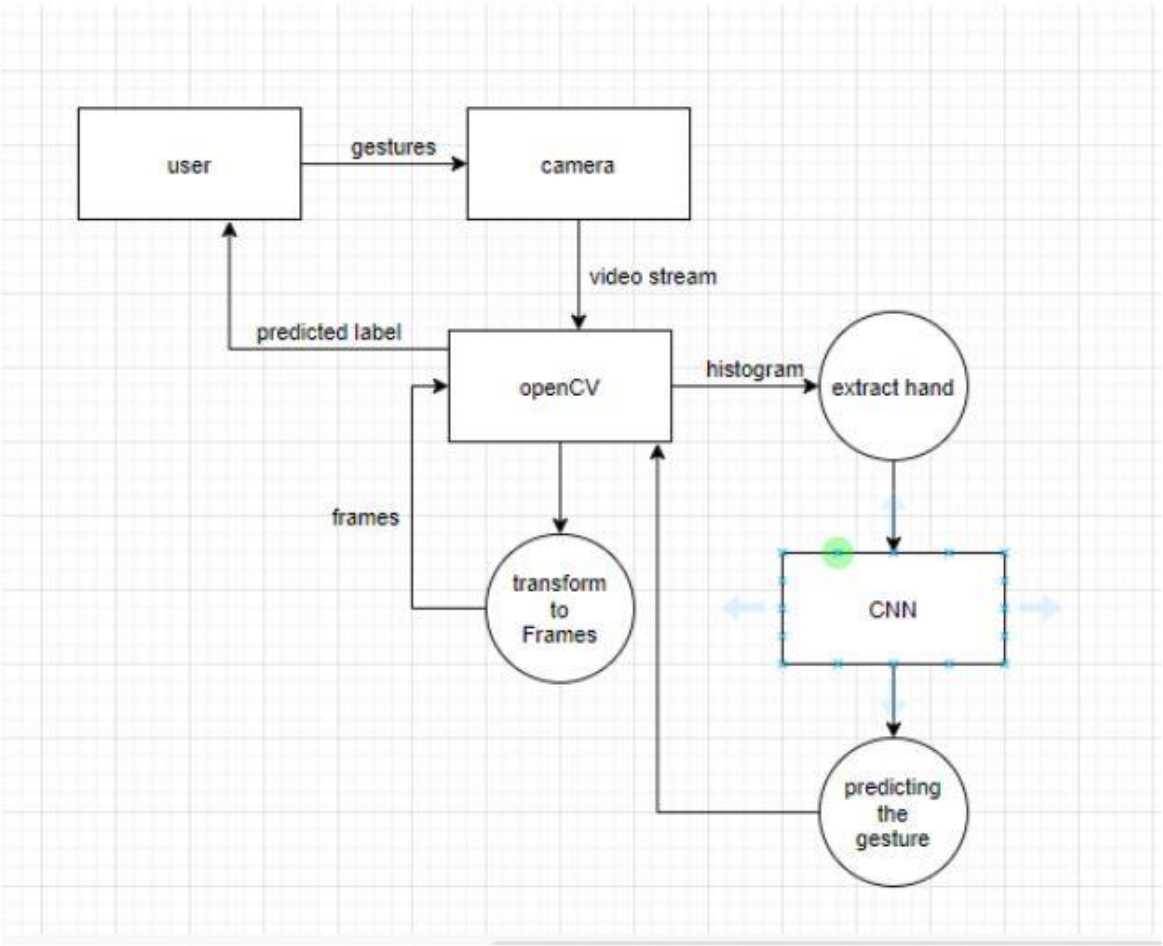


Fig . 4.2

4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. Taking SRS document of analysis as input to the design phase drawn UML diagrams. The UML is only language so is just one part of the software development method. The UML is process independent, although optimally it should be used in a process that should be

driven, architecture-centric, iterative, and incremental. The UML is language for visualizing, specifying, constructing, documenting the articles in a software-intensive system. It is based on diagrammatic representations of software components. A modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of the system. A modeling language such as the UML is thus a standard language for software blueprints. The UML is a graphical language, which consists of all interesting systems. There are also different structures that can transcend what can be represented in a programming language.These are different diagrams in UML

4.3.1 USE CASE DIAGRAMS

Use Case during requirement elicitation and analysis to represent the functionality of the system. Use case describes a function by the system that yields a visible result for an actor. The identification of actors and use cases result in the definitions of the boundary of the system i.e., differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system. Use case describes the behaviour of the system as seen from the actor's point of view. It describes the function provided by the system as a set of events that yield a visible result for the actor.

Purpose of Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

ln brief, the purposes of use case diagrams can be said to be as follows −

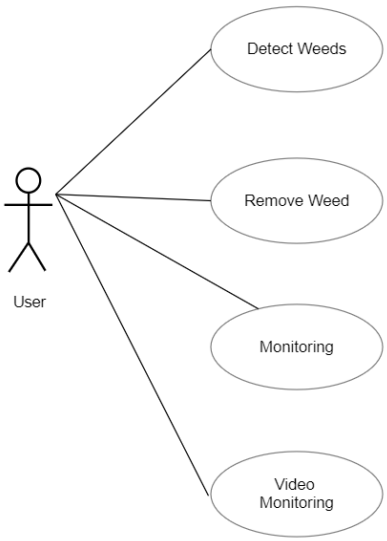
- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.
- How to Draw a Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases. We can say that use cases are nothing, but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system. Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

Functionalities to be represented as use case Actors Relationships among the use cases and actors. Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we must use the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important.
- The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig. 4.3


Detect Weeds
Remove Weed
Monitoring
Video
Monitoring
User

4.3.2 CLASS DIAGRAM

Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagram describe the different perspective when designing a system-conceptual, specification and implementation. Classes are composed of three things: name, attributes, and operations. Class diagram also display relationships such as containment, inheritance, association etc. The association relationship is most common relationship in a class diagram. The association shows the relationship between instances of classes.

How to Draw a Class Diagram?

Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram. Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top level view.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represent the whole system.

The following points should be remembered while drawing a class diagram −

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified for each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing it should be understandable to the developer/coder.
- Finally, before making the final version, the diagram should be drawn on

plain paper and reworked as many times as possible to make it correct.

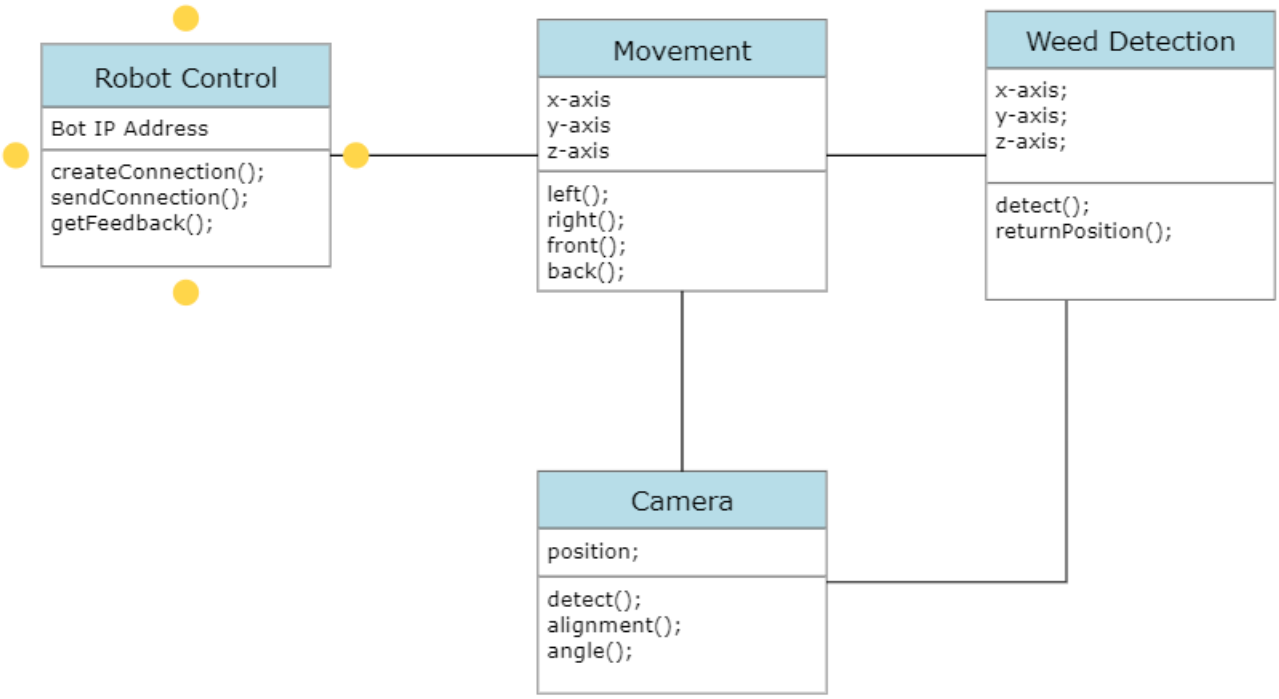


Fig. 4.4

4.3.3 SEQUENCE DIAGRAM

Sequence diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension(time) and horizontal dimension (different objects).

Objects: Object can be viewed as an entity at a particular point in time with specific value and as a holder of identity.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence

diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances. Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn't have to wait for a response. Asynchronous calls are present in multithreaded applications, event-driven applications and in message-oriented middleware. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message (ExecutionSpecifications in UML). Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. If an object is destroyed (removed from memory), an X is drawn on bottom of the lifeline, and the dashed line ceases to be drawn below it. It should be the result of a message, either from the object itself, or another. A message sent from outside the diagram can be represented by a message originating from a filled-in circle (found message in UML) or from a border of the sequence diagram (gate in UML). UML has introduced significant improvements to the capabilities of sequence diagrams. Most of these improvements are based on the idea of interaction fragments which represent smaller pieces of an enclosing interaction. Multiple interaction fragments are combined to create a variety of combined fragments, which are then used to model interactions that include parallelism, conditional branches,optional interactions

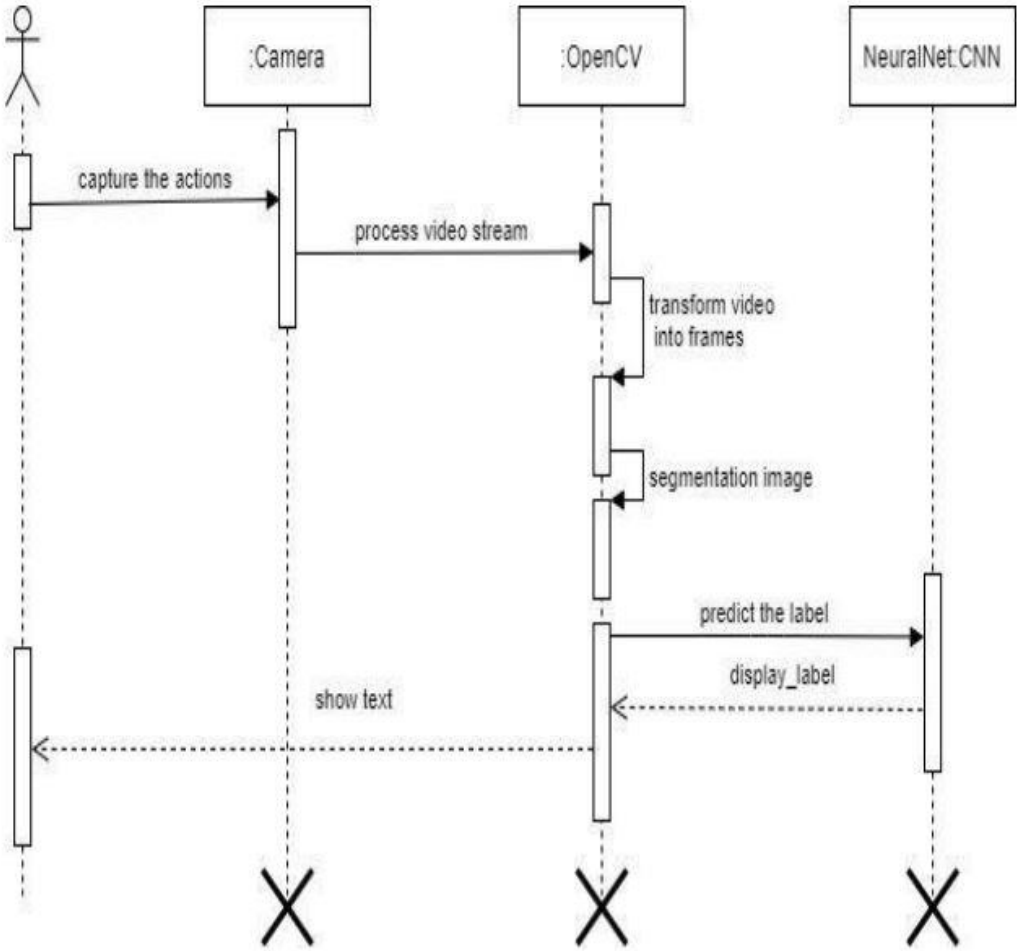


Fig. 4.5

4.3.4 STATE CHART DIAGRAM

A state chart diagram describes a state machine which shows the behaviour of classes. It shows the actual changes in state not processes or commands that create those changes and is the dynamic behaviour of objects over time by model1ing the life cycle of objects of each class. It describes how an object is changing from one state to another state. There are mainly two states in State Chart Diagram:

1. Initial State
2. Final-State.

Some of the components of State Chart Diagram are:

State: It is a condition or situation in life cycle of an object during which it's satisfies ame condition or performs some activity or waits for some event.

Transition: It is a relationship between two states indicating that object in first state performs some actions and enters into the next state or event.

Event: An event is specification of significant occurrence that has a location in time

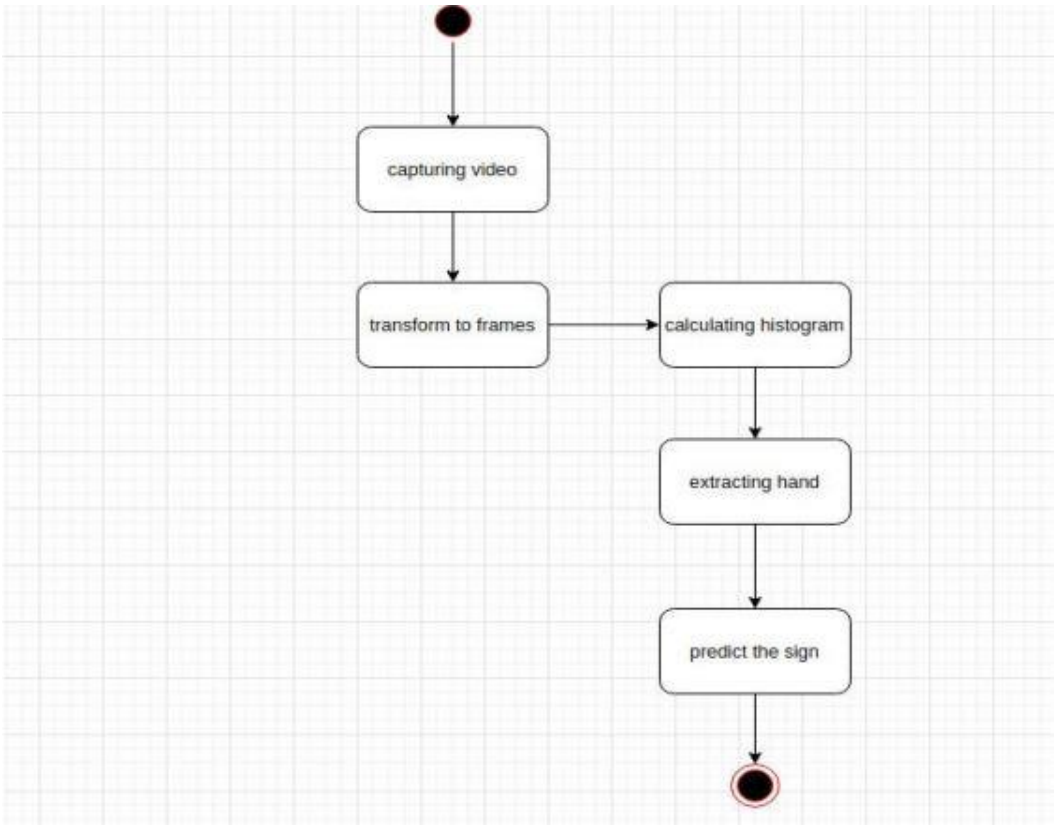


Fig. 4.6

CHAPTER 5

MODULE DESCRIPTION

CHAPTER 5

MODULE DESCRIPTION

5.1 ARDUINO SOFTWARE MODULE(IDE)

- Writing Sketches
- File
- Edit
- Sketch
- Tools
- Help
- Sketchbook
- Tabs, Multiple Files, and Compilation
- Uploading
- Libraries
- Third-Party Hardware
- Serial Monitor
- Preferences
- Language Support
- Boards

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.

5.2 PYTHON MODULE

Python is an interpreted high-level programming language for programming Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

PYTHON FEATURES:

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented , imperative, functional and procedural, and has a large and comprehensive library. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect - oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

PYTHON LIBRARIES

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing

5.3 OPENCV MODULE

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. Numpy is a highly optimized

library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

Since OpenCV is an open source initiative, all are welcome to make contributions to this library. And it is same for this tutorial also. So, if you find any mistake in this tutorial (whether it be a small spelling mistake or a big error in code or concepts, whatever), feel free to correct it

And that will be a good task for freshers who begin to contribute to open source projects. Just fork the OpenCV in github, make necessary corrections and send a pull request to OpenCV. OpenCV developers will check your pull request, give you important feedback and once it passes the approval of the reviewer, it will be merged to OpenCV. Then you become a open source contributor. Similar is the case with other tutorials, documentation etc. As new modules are added to OpenCV-Python, this tutorial will have to be expanded. So those who knows about particular algorithm can write up a tutorial which includes a basic theory of the algorithm and a code showing basic usage of the algorithm and submit it to OpenCV. Remember, we together can make this project a great success !!!

CHAPTER 6

TESTING

CHAPTER 6
TESTING

Discovering and fixing such problems is what testing is all about. The purpose of testing is to find and correct any problems with the final product. It's a method for evaluating the quality of the operation of anything from a whole product to a single component. The goal of stress testing software is to verify that it retains its original functionality under extreme circumstances. There are several different tests from which to pick. Many tests are available since there is such a vast range of assessment options.

Who Performs the Testing: All individuals who play an integral role in the software development process are responsible for performing the testing. Testing the software is the responsibility of a wide variety of specialists, including the End Users, Project Manager, Software Tester, and Software Developer.

When it is recommended that testing begin: Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase. In the waterfall model, the phase of testing is where testing is explicitly arranged and carried out. Testing in the incremental model is carried out at the conclusion of each increment or iteration, and the entire application is examined in the final test.

When it is appropriate to halt testing: Testing the programme is an ongoing activity that will never end. Without first putting the software through its paces, it is impossible for anyone to guarantee that it is completely devoid of errors. Because the domain to which the input belongs is so expansive .

6.1 TYPES OF TESTING

There are four types of testing:

Unit Testing

The term "unit testing" refers to a specific kind of software testing in which discrete elements of a program are investigated. The purpose of this testing is to ensure that the software operates as expected.

Test Cases

1. Test that the YOLO model can correctly detect and classify different types of accidents, such as car crashes, pedestrian accidents, or bicycle accidents.
2. Test that the YOLO model can accurately detect accidents in different lighting conditions, such as daytime, nighttime, or low-light conditions.
3. Test that the YOLO model can correctly identify accidents in different weather conditions, such as rain, snow, or fog.

Integration Testing

The programme is put through its paces in its final form, once all its parts have been combined, during the integration testing phase. At this phase, we look for places where interactions between components might cause problems.

Test Cases

1. Test the integration between the YOLO model and the data pre-processing pipeline to ensure that the input data is properly formatted and pre-processed before being fed into the model.
2. Test the integration between the YOLO model and the post-processing pipeline to ensure that the output predictions are properly processed and formatted before being sent to the user interface or other downstream systems.

3. Test the integration between the YOLO model and the user interface to ensure that the system is able to receive user input and display the output predictions in a user-friendly and intuitive manner.

Functional Testing

One kind of software testing is called functional testing, and it involves comparing the system to the functional requirements and specifications. In order to test functions, their input must first be provided, and then the output must be examined. Functional testing verifies that an application successfully satisfies all of its requirements in the correct manner. This particular kind of testing is not concerned with the manner in which processing takes place; rather, it focuses on the outcomes of processing. Therefore, it endeavours to carry out the test cases, compare the outcomes, and validate the correctness of the results.

Test Cases

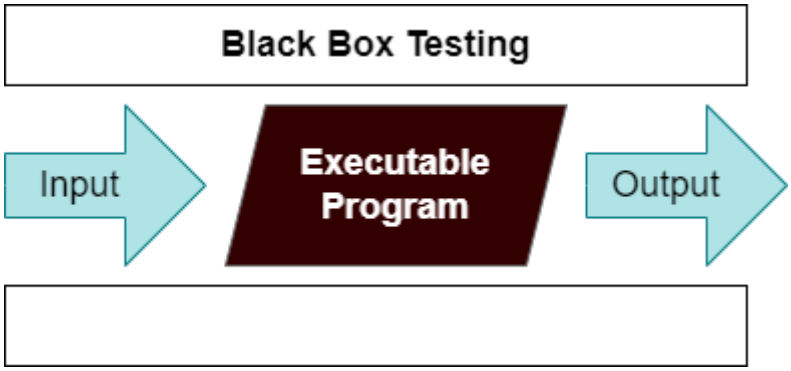
1. Test that the system can accurately detect accidents in real-time scenarios, using a variety of test videos and images that simulate different types of accidents.
2. Test that the system can accurately classify the type of accident, such as car crashes, pedestrian accidents, or bicycle accidents, and generate appropriate notifications or alerts.
3. Test that the system can accurately localize the accident within the video or image, such as identifying the exact location of the impact or collision.

6.2 TESTING TECHNIQUES

There are many different techniques or methods for testing the software, including the following:

BLACK BOX TESTING

During this kind of testing, the user does not have access to or knowledge of the internal structure or specifics of the data item being tested. In this method, test cases are generated or designed only based on the input and output values, and prior knowledge of either the design or the code is not necessary. The testers are just conscious of knowing about what is thought to be able to do, but they do not know how it is able to do it.



For example, without having any knowledge of the inner workings of the website, we test the web pages by using a browser, then we authorise the input, and last, we test and validate the outputs against the intended result.

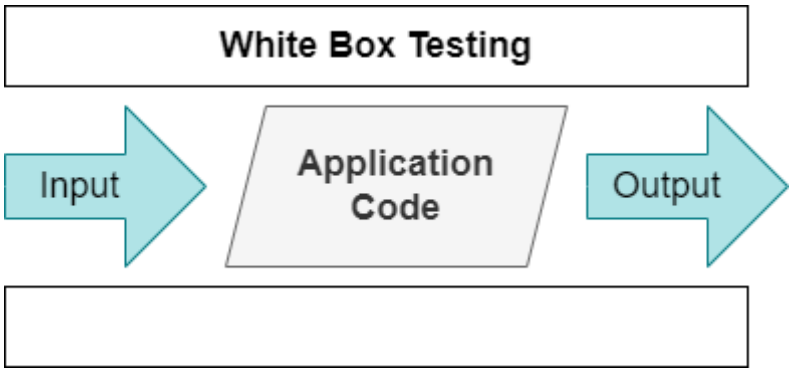
Test Cases

1. Test the input data quality and boundary conditions, such as the resolution, aspect ratio, color format, and file format of the input images or videos.
2. Test the model's ability to detect and classify accidents under various environmental conditions, such as different weather conditions, lighting conditions, and camera angles.
3. Test the model's accuracy and sensitivity to different types of accidents, such as car crashes, pedestrian accidents, or bicycle accidents.

WHITE BOX TESTING

During this kind of testing, the user is aware of the internal structure and details of the data item, or they have access to such information. In this process, test

cases are constructed by referring to the code. Programming is extremely knowledgeable of the manner in which the application of knowledge is significant. White Box Testing is so called because, as we all know, in the tester's eyes it appears to be a white box, and on the inside, everyone can see clearly. This is how the testing got its name.



As an instance, a tester and a developer examine the code that is implemented in each field of a website, determine which inputs are acceptable and which are not, and then check the output to ensure it produces the desired result. In addition, the decision is reached by analyzing the code that is really used.

Test Cases

1. Test the correctness of the YOLO model's architecture, such as ensuring that the correct number and types of layers are present, and that the correct parameters and hyperparameters are used.
2. Test the correctness of the training process, such as ensuring that the training data is properly preprocessed, and that the model is trained with the correct loss function and optimizer.

CHAPER 7

CONCLUSION

CHAPER 7

CONCLUSION

7.1 CONCLUSION

In conclusion, the Gesture Controlled Wireless Agricultural Weeding Robot is a highly advanced and innovative technology that has the potential to revolutionize the agricultural industry. In this project, we have developed a Trainable automatic robot which helps in removing unwanted weed on agricultural fields using gesture to control a three-axis robotic arm to do the necessary work. The arm is placed on a rover which is controlled Wirelessly using Bluetooth. The arm is taught to do the repetitive motion with a gesture using a hand glove to do the necessary work. The setup of the rover with attached the robotic arm is tested and evaluation under normal environmental conditions This robot allows farmers to automate the weeding process, which can save a significant amount of time, effort, and money.

CHAPTER 8
APPENDIX 2

CHAPTER 8

APPENDIX 2

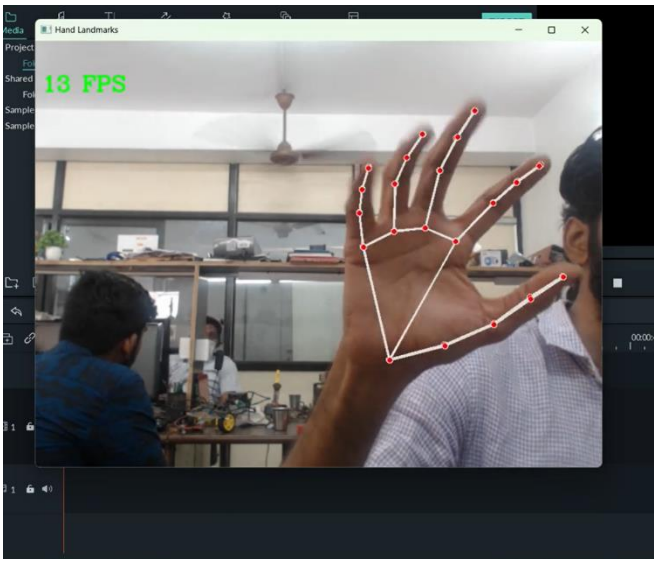


Fig 8.1

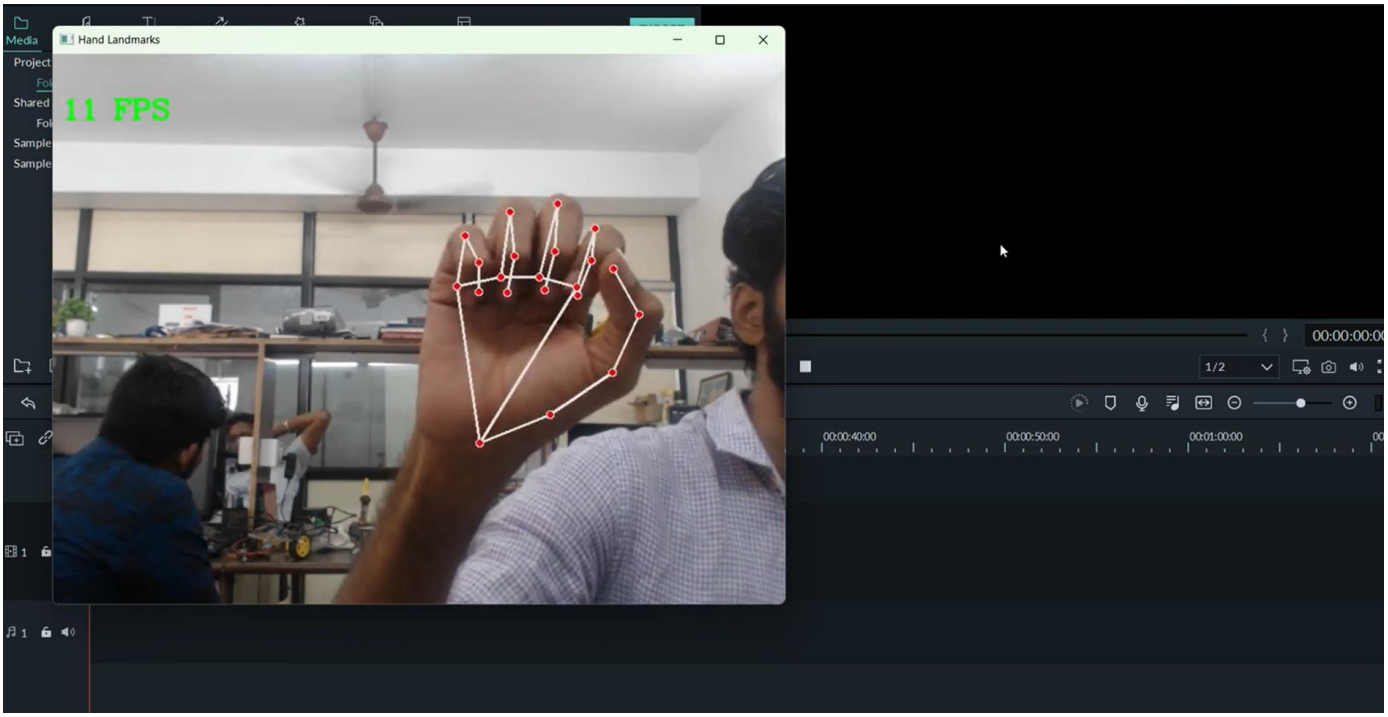


Fig 8.2

CHAPTER 9

REFERENCES

CHAPTER 9

REFERENCES

[1] Nasser H. Dardas and Nicolas D. Georganas, “Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques”, IEEE transactions on instrumentation and measurement, vol. 60, no. 11, November 2011.

[2] L. Yun and Z. Peng, “An automatic hand gesture recognition system based on Viola-Jones method and SVMs,” in Proc. 2nd Int. Workshop Comput. Sci. Eng., 2009, pp. 72–76.

[3] W. Chung, X. Wu, and Y. Xu, “A real time hand gesturerecognition based on Haar wavelet representation,” in Proc. IEEE Int. Conf. Robot. Biomimetics, 2009, pp. 336–341.

[4] L. Juan and O. Gwun, “A comparison of SIFT, PCA-SIFT and SURF,” Int. J. Image Process. (IJIP), vol. 3, no. 4, pp. 143–152, 2009.

[5] L. Yeffet and L. Wolf, “Local trinary patterns for human action recognition,” in Proc. IEEE ICCV, 2009, pp. 492–497.

[6] Y. Ren and C. Gu, “Real-time hand gesture recognition based on vision,” in Proc. Edutainment, 2010, pp. 468–475