Chapter 1

Introduction

Vietnam is on the fast development track. However, the traditional culture is always treasured. The concept of Automation of Nem-Con is based on an interesting folk game called "Shuttlecock Throwing.

Figure 1.1: Nem-Con Festival in Vietnam

Figure 1.2 Shuttlecock

A bamboo ring is hung on top of the bamboo stick. The ring is covered in yellow and pink papers. Yellow color represents the moon, while pink represents the sun. The heart and soul of this game is the Shuttlecock. The shuttlecock is an object made of cotton balls or filled with rice husks, symbolizing prosperity and happiness. The shuttlecock is hung by a string sewn to the center of the Shuttlecock. It is decorated with colorful cotton representing the colors of the rainbow. When playing, the player holds the end of the string to swing it clockwise various times before throwing and aims for the center of the ring. If the shuttlecock makes it through the ring, the player wins.

Figure 1.3 Rings

Figure 1.4 Rings

The flying shuttlecock depicts a flying dragon, iconic of human power and the universe. That's why the Shuttlecock Throwing Festival opening is commenced with a ceremony to pray for deities of the Land and Sky. After the first shuttlecock hits the target, it will be opened. The husks inside it are shared among the people as a wish for a prosperous year. Nobody knows when the game started. Today, it is not only a folk game for both men and women, where they can meet and find their other halves but also a sport for players to show their skills.

1.1 Problem Definition

To build a manual and automated robot that aims and throws a shuttle cock through the ring and land in a specified area.

1.2 Justification of Problem

- While democratization of robots has become more and more important the presence of robots in our daily life has become more essential.
- Robots tend to push the boundaries of biology, cognitive science, and engineering, generating a mountain of scientific publications in many fields related to humans and make their lives easier.
- Hence building robots helps to save time and improve efficiency which otherwise cannot be attained by human intervention.

1.3 Need of Problem

- During the building of the robot in the initial stage several papers were referred from various conferences majorly from IEEE, IJETT, International Journal of Advanced Engineering Research and Studies and Journal of Engineering Science and Technology.
- The current work relates to several streams of work in building of robots for industrial purpose. We review them selectively in this section to provide a context for this work.

Review and Description of Papers Referred:

1) A Survey Paper on PID Control System: This paper concentrated on the work done on the PID control system specially on time delaying systems like network control systems. The different works which have already been done are summarized like gain margin, phase margin methods. The absolute focal point on time delayed system are the delays induced by network and the packet losses. This survey paper covers the study on such methods that compensates these terms.

2) <u>PID Control System Analysis, Design, and Technology</u>: Designing and tuning a proportional-integral-derivative (PID) controller appears to be conceptually intuitive, but can be hard in practice, if multiple (and often conflicting) objectives such as short transient and high stability are to be achieved. Usually, initial designs obtained by all means need to be adjusted repeatedly through computer simulations until the closed-loop system performs or compromises as desired.

3) <u>Study Of Tuning Of PID Controller By Using Particle Swarm Optimization</u>: Particle Swarm Optimization (PSO) algorithm approach to generate the optimal tuning parameters. The paper dealt with optimal tuning of proportional integral derivative (PID) controller for controlling the output obtained and hence to minimize the integral of absolute errors. The main objective was to obtain a stable, robust and controlled system by tuning the PID controller using Particle Swarm Optimization (PSO) algorithm. It is necessary to use PID controller to increase the stability and performance of the system. Fast tuning of optimum PID controller parameter yield high quality solution. The paper demonstrated in detail how to employ the PSO method to search efficiently the optimal PID controller parameters.

Literature Survey

Sr. no	Title	Author, Date	Rewards
1	Accurate Object Throwing by an Industrial Robot Manipulator	Wilhelm August, Steffen Waeldele, Bjoern Hein and Heinz Woern	1)Experimental setup of industrial robot. 2)Path planning Algorithm
2	PID Control System Analysis, Design, and Technology	Kiam Heong Ang, Gregory Chong, Student Member, IEEE, and Yun Li, Member, IEEE IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 13, NO. 4, JULY 2005	1)Proportional-integral-derivative (PID) control 2)PID hardware, PID software, PID tuning.
3	A Survey Paper on PID Control System	Daya Sagar Sahu, Sunil Sharma International Journal of Engineering Trends and Technology (IJETT) – Volume 21 Number 7 – March 2015	1)proportional Control Systems 2)Tuning Parameters and common methods used
4	Study Of tuning Of PID Controller By Using Particle Swarm Optimization	Ankita Nayak1 , Mahesh Singh M.E. Scholar, 2 Sr. Asstt. Professor, SSGI, SSTC India International Journal of Advanced Engineering Research and Studies	1)Design Of PID controller. 2)Optimization technique POS: Particle Swarm Optimization

Table 1.1: Literature Survey Paper

1.4 Purpose of the system

- A primary goal of the project is to advance the state of robotic working which is currently limited in many cases to one robotic system controlled by a single controller.
- The project will approach this problem not only at the robot level, but also by developing a higher-level framework for analysing the whole process, including the roles and interactions of both humans (manual robot) and robots(automatic).
- Developing a higher-level framework for the roles and interactions of both humans (manual robot) and robots(automatic). To develop a comprehensive system to optimize and enhance the intelligence of the robot.
- Development and analysis of algorithms to learn and perform the activities with minimal human intervention is the areas which we will explore in the process of building the robot.
- Building a manual robot (MR) operated by operator via wireless or cable connection and an automatic robot (AR) which will work without any help from an operator.

Chapter 2

Analysis

Analysis of the system means estimating the cost and time required for project creation and analyzing the requirements from the customer. Project cost estimate calculates cost required and time estimate calculates the time required for building the system as shown in figure 2.1.1 Requirements analysis is a software engineering task that bridges the gap between system level requirements engineering and software design. It also provides the software designer with a representation of information and function of the system that can be used to create design.

2.1 Project Plan

2.1.1 Project Estimates

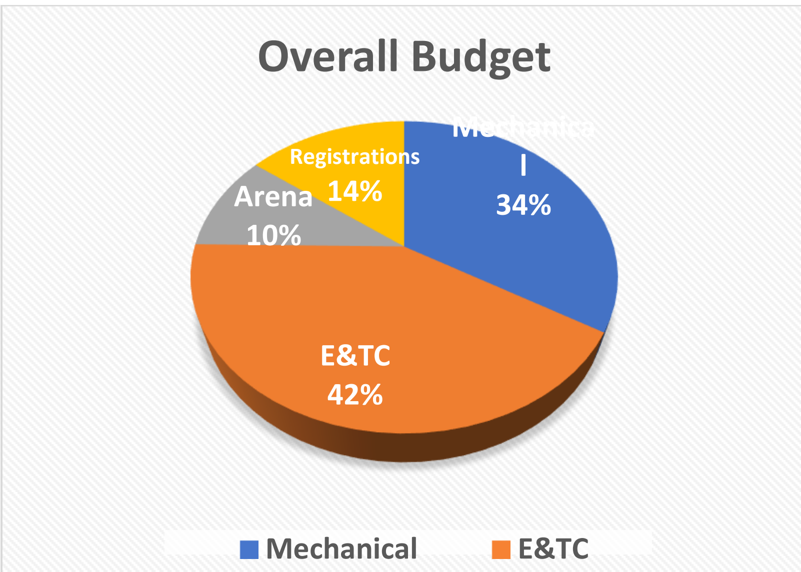


Figure 2.1.1 Budget

2.1.2 Reconciled Estimates

SR. NO	DEPARTMENT	BUDGET
1)	MECHANICAL	198550
2)	E&TC	215845
3)	TEAM REGISTRATION	25000
3)	INDIVIDUAL REGISTRATIONS (1750*35)	61250
4)	ARENA	60000
	GRAND TOTAL	**560645**

Table 2.1.2 Reconciled Estimates

2.1.2.1 Cost Estimate

ROBOCON 2018 Budget

SR. NO	COMPONENT	RATE	QUANTITY	AMOUNT
A) CHASSIS (MANUAL + AUTOMATIC ROBOT)				
1)	ALUMINIUM 6075 SQUARE SECTIONS (12*12 mm)		40FT.	1500
2)	OTHER SECTIONS (‘L’ AND ‘C’) FOR SUPPORT		24FT.	1000
3)	ALUMINIUM BILLET 6082 FOR WHEEL FLANGE (DIA. 50*500mm)	950/KG	2	8000
4)	ALUMINIUM PLATE FOR MOTOR MOUNTING (500*500*3 mm)	550/KG	2	1500
5)	MACHINING AND WELDING			15000
6)	CAST NYLON (500*400*12mm)	2250	2	4500

7)	OMNI WHEELS	5500	4	22000
8)	BUSH COUPLINGS (MICKEY PULLEYS)	750	8	6000
9)	ACRYLIC PLATE FOR LSA-08 MOUNTING WITH LASER CUTTING (400*400*1mm)			1000
10)	BALL BEARINGS	200	8	1600
B) THROWIN G MECHANISM				
11)	PNEUMATIC GRIPPER	11500	2	23000
12)	PNEUMATIC CYLINDER (BORE DIA. 16mm)	3000	2	6000
13)	DIRECTION CONTROL VALVES (SOLENOID OPERATED)	1500	4	6000
14)	AIR RESERVOIR	1000	2	2000
15)	MALE CONNECTOR	50	20	1000
16)	PRESSURE GAUGE	500	2	1000

17)	HOSES (DIA.-3mm, 5mm, 10mm)			1000
18)	MUFFLER	200	2	400
19)	QUICK EXHAUST VALVE	300	1	300
20)	FLOW CONTROL VALVE	150	4	600
21)	ELECTRONIC PRESSURE REGULATOR	30000	1	30000
22)	ARALDITE, TEFLON			500
23)	ALUMINIUM BILLET FOR MOUNTING			5000
24)	STEPPER MOTOR MOUNTINGS AND COUPLINGS			8000
25)	SHUTTLECOCKS	300	15	4500
26)	MACHINING (WATER JET+ MILLING)			15000
27)	BEARINGS	200	2	400
C) MANUAL ROBOT MEC HANISM				
28)	RACK FOR SHUTTLECOCKS			1000
29)	ELECTRONIC GRIPPERS (ANGULAR) GRIPPING FORCE = 40N	1000	2	2000

30)	CAST NYLON FOR RACK AND PINION			1500
31)	PP PLATES MALE AND FEMALE PARTS OF PICKING MECHANISM			1500
32)	MACHINING			7500
33)	STEPPER MOTOR COUPLING	750	1	750
34)	HOOKS	50	10	500
35)	MECHANICAL ACCESSORIES (NUT BOLT ETC)			5000
36)	LINEAR ACTUATORS (200 mm STROKE)	2	6000	12000
			TOTAL	**198550**
	D) ARENA			**60000**
			GRAND TOTAL	**258550**

Table 2.1.2.1 Cost Estimates

SR. NO	COMPONENT	RATE	QTY	AMOUNT
A) MAJOR COMPONENTS				
1)	ORANGE 16000MAH 6S 25/50C LI PO BATTERY	22869	3	68607
2)	7.2 KG CM STEPPER MOTOR	1120	12	16800
3)	4.5AMP BIPOLAR STEPPING DRIVER	3950	12	47400
4)	CYTRON DUAL CHANNEL ENHANCED 13AMP DRIVER	1890	8	15120
5)	5-42V BOOST CONVERTERS	350	15	5250
6)	ARDUINO DUE BOARD	2450	4	9800
7)	ARDUINO NANO BOARD	1199	10	11990
8)	ADVANCED AUTO CALIBRATING LINE SENSOR LSA08	4725	5	23625
9)	SKYRC IMAX B6AC V2 PROFESSIONAL BALANCE CHARGER/DISCHARGER(ORIGINAL)	5489	1	7489

10)	PCB	1500	2	3000
B) BASIC COMPONENTS				
11)	MULTITEC 150B WIRE STRIPPER	57	4	228
12)	MULTITEC 07 NIPPER	94	2	188
13)	DOUBLE TAPE	40	4	160
14)	LI PO VOLTAGE CHECKER	199	4	796
15)	DIGITAL MULTIMETER	300	2	600
16)	SOLDRON 100W SOLDERING IRON	651	2	1302
17)	SOLDERING METAL ROLL 50GMS 22 GAUGE SN60/PB40	103	4	412
18)	GREEN MASKING SOLUTION	80	2	160
19)	INSULATION TAPE	20	4	80
20)	PAPER TAPE	40	4	160
21)	ALL PURPOSE HOT METAL GLUE STICK FOR GLUE GUN	68	5	68
22)	STANDARD TEMPERATURE CORDED 150MM 40W HOT METAL GLUE GUN	259	2	518

23)	USB CABLES	51	2	102
24)	FRC CABLES	51	4	204
25)	10CM MALE TO MALE BREADBOARD JUMPER DUPONT 1P-1P CABLE 40 PCS	60	2	120
26)	10CM MALE TO FEMALE BREADBOARD JUMPER DUPONT 1P-1P CABLE 40 PCS	60	2	120
27)	10CM FEMALE TO FEMALE BREADBOARD JUMPER DUPONT CABLE 40 PCS	60	2	120
28)	COMBO DRILL BIT FOR PCB	250	1	250
29)	SCREW DRIVER	22	4	88
30)	BATERRY CONNECTOR	114	4	456
31)	HEAT SHRINKS	7	20	140
32)	EMERGENCY SWITCH	99	2	192
33)	ETCHING SOLUTION	300	1	300
		GRAND TOTAL		**215845**

Table 2.1.2.1 Cost Estimates

2.1.2.2 Time Estimates

The building of the robot started about a year back after the declaration of the theme by the ROBOCON club. Date: 21st July 2017. The robot of Robocon 2017 theme was displayed and introduced to students of all the classes of all the departments. Various workshops were conducted for the on ARDUINO and basic robotics. The time estimate to complete the project and build the robot was about a year.

Sr. No.	Activity	Estimated Hours(Hrs.)
1	Coding the manual Robot	35
2	Coding the Automated Robot	75
3	Building the Chassi Design	55
4	Building the manual Robot	55
5	Building the Automated Robot	55
6	Total time estimate	Total: 275 hrs.

Table 2.1.2.2 Time Estimates

2.1.3 Project Resources

Project resources: Various hardware components were used along with tools and software. Hardware components like stepper motors, LSA controllers, Emergency switches, Arduino Due Board and were the most prominent hardware devices that were used.

2.2 Mathematical Modelling

Let R be the robot that describes the model P, hence P can be given as

$$P= \{s, e, x, y, DD, NDD, Fme\_shared\}$$

Where,

1. S=Start State
   - The manual robot and the automated robot are outside the play area in the Start Zone.
2. E=End State
   - When Golden Shuttlecock is thrown through the Golden Ring and then is landed on the Golden Cup, that team will gain the victory and the match will be finished. This victory is called “Rongbay” (“Flying Dragon”) which is the end state from the final zone.
3. X=Input State
   - The co-ordinates and the dimensions of the play zone and the field is the input that will be given to the robot.
4. Y=Output state
   - Calculating and throwing the shuttle-cock is the action performed by the robot.
5. DD=Deterministic data
6. NDD=Non-deterministic data
7. Fme_shared=Memory shared by the processors (ARM-Cortex) to evaluate and process the data

2.2.1 Goals and objectives

- A robot is a droid or a mechanical device which is capable of executing one or more tasks automatically with speed and precision.
- Our main motivation behind building this robot is to achieve highest level of accuracy and precision in executing the expected task.
- The robots of tomorrow are expected to play an active role in the real world inhibited by objects of uncertainty.
- As the, project is in the domain of smart systems, the robot will be artificially intelligent and will have the intelligence for decision making and analytical reasoning for the particular circumstance.
- E.g. The robot will have the intelligence to make the decision of whether or not the shuttlecock has passed through the ring and to make the next throw through the same ring or to proceed to the next zone.
- Developing a higher-level framework for the roles and interactions of both humans (manual robot) and robots(automatic).
- Achieving highest level of accuracy by throwing the shuttlecock and improve the chances of it passing through the ring and precision in executing the expected task.

Objectives

- A primary goal of the project is to advance the state of robotic working which is currently limited in many cases to one robotic system controlled by a single controller.
- The project will approach this problem not only at the robot level, but also by developing a higher-level framework for analysing the whole process, including the roles and interactions of both humans (manual robot) and robots(automatic).
- To obtain accuracy to throw the shuttlecock via a robot and improve the chances of it passing through the ring which is not possible via a human being.
- Other objective is to obtain sustainability and communication between the manual and the automatic robot.
- To develop a comprehensive system to optimize and enhance the intelligence of the robot.
- To develop analytical reasoning skills for the particular circumstance and make appropriate decisions.

- Investigate and understand various mechanisms and determine the purpose of application.

2.2.2 Statement of Scope

- Building a manual robot (MR) operated by operator via wireless or cable connection and an automatic robot (AR) which will work without any help from an operator.
- The manual robot will give the shuttlecock to the automatic robot and then the automatic robot will throw it through the ring.
- It mainly focuses on building a smart system which artificially intelligent robot which will be able to make the decisions on its own.
- It will be capable of making the decision of throwing the shuttle-cock through the ring with precision and changing the zones (play area).
- Development and analysis of algorithms to learn and perform the activities with minimal human intervention is the areas which we will explore in the process of building the robot.

2.2.3 Methodologies of Problem solving and efficiency issues

• Other technologies such as, Mouser Electronics and a collection of products, articles, and resources especially suited to making robots and related robotics can be used for building the robot.

• Deep learning algorithms can for improving the AI of the system also could be used for line tracing and decision making of the robot but it would increase the complexity of the project.

2.2.4 Major Constraints

- As the robot uses artificial intelligence for the passing of the shuttle-cock to the AR (Automatic robot). This is where the major challenge for the robot is.
- Also, the weight of the robot should not exceed 25kgs, which makes it even more difficult for the user to use the use high power battery packed motors and drivers which reduces the power of the robot.

2.2.5 Outcome

- As the, project is in the domain of smart systems, the robot is artificially intelligent and uses it for decision making and analytical reasoning for the particular circumstance. E.g. The robot uses artificial intelligence to make the decision of whether or not the shuttlecock has passed through the ring and to make the next throw through the same ring or to proceed to the next zone.

- The outcome is to build two robots a manual and an automatic robot the manual robot is operated by operator via wireless or cable connection(MR). The manual robot gives the shuttlecock to the automatic robot and then the automatic robot throws it through the ring. The automatic robot is able to work independently without any help from an operator(AR)
- The robot accurately throws the shuttlecock via a robot and improve the chances of it passing through the ring which is not possible via a human being. Other objective is to obtain sustainability and communication between the manual and the automatic robot.
- A primary goal of the project is to advance the state of robotic working which is currently limited in many cases to one robotic system controlled by a single controller.
- The project will approach this problem not only at the robot level, but also by developing a higher-level framework for analysing the whole process, including the roles and interactions of both humans (manual robot) and robots(automatic).
- To develop a comprehensive system to optimize and enhance the intelligence of the robot. To develop analytical reasoning skills for the particular circumstance and make appropriate decisions. Investigate and understand various mechanisms and determine the purpose of application.

2.2.6 Applications

- The robot can be used for national level competitions for throwing mechanism. Also, as it uses PID controller and PID algorithm it can be used for line tracing.
- It is basically a robot which follows a particular path or trajectory and decides its own course of action which interacts with obstacle.

2.3 Requirement Analysis

The requirement analysis of a project means to determine whether the needs of the user are satisfied by the specific software or not. In this project the requirement analysis will be a wether or not the shuttle-cock is successfully thrown from the ring and also efficient and workable. The requirements must maintain a standard quality of its requirement, different types of requirement quality includes:

- Complete
- Testable
- Unambiguous
- manipulative
- amount human-robot interaction
- sensing
- power/endurance

2.4 Advances/additions/updating the previous system

- Precision in tracing the line and throwing action of the robot to successfully pass through the ring.

2.5 Team Organization

- **Team Structure**

The team consists of two people from the computer department Rajarshri Godse-Head and student Co-Ordinator for ROBOCON team and Aishwarya Murkute

Aishwarya Murkute	Programming (Manual robot and Automated robot) and MATLAB
Rajarshri Godse	Programming (Automatic Robot) MATLAB and interfacing

Chapter 3

Design of the Robot

This chapter gives a brief about the system architecture along with its software requirements, features, non-functional features, performance and safety requirements. This chapter also give the information on the interfacing between different software used in the system. The impact of different kinds of risks on the system is also specified in this chapter.

3.1 Software Requirement Specification

3.1.1 (I) Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Fig. 3.6) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

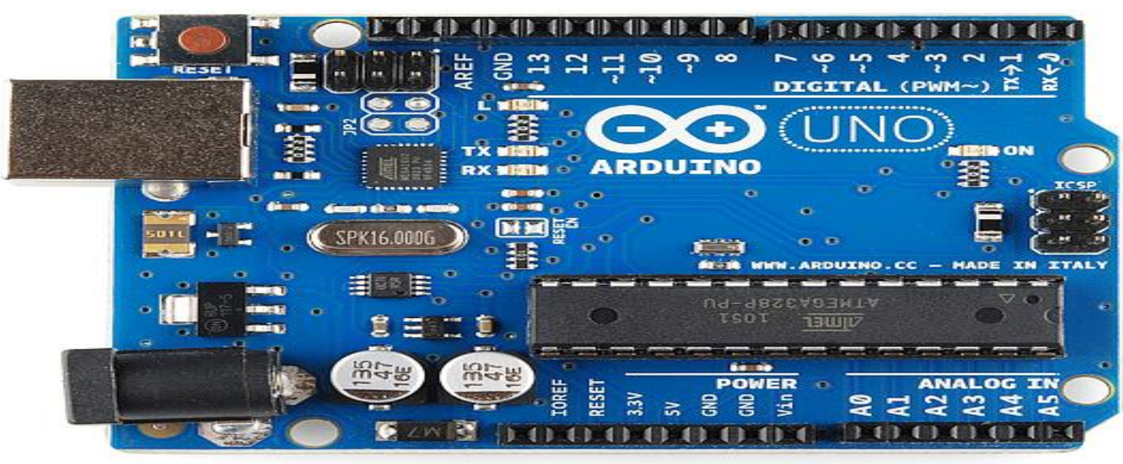


Figure 3.1.1 Arduino Uno

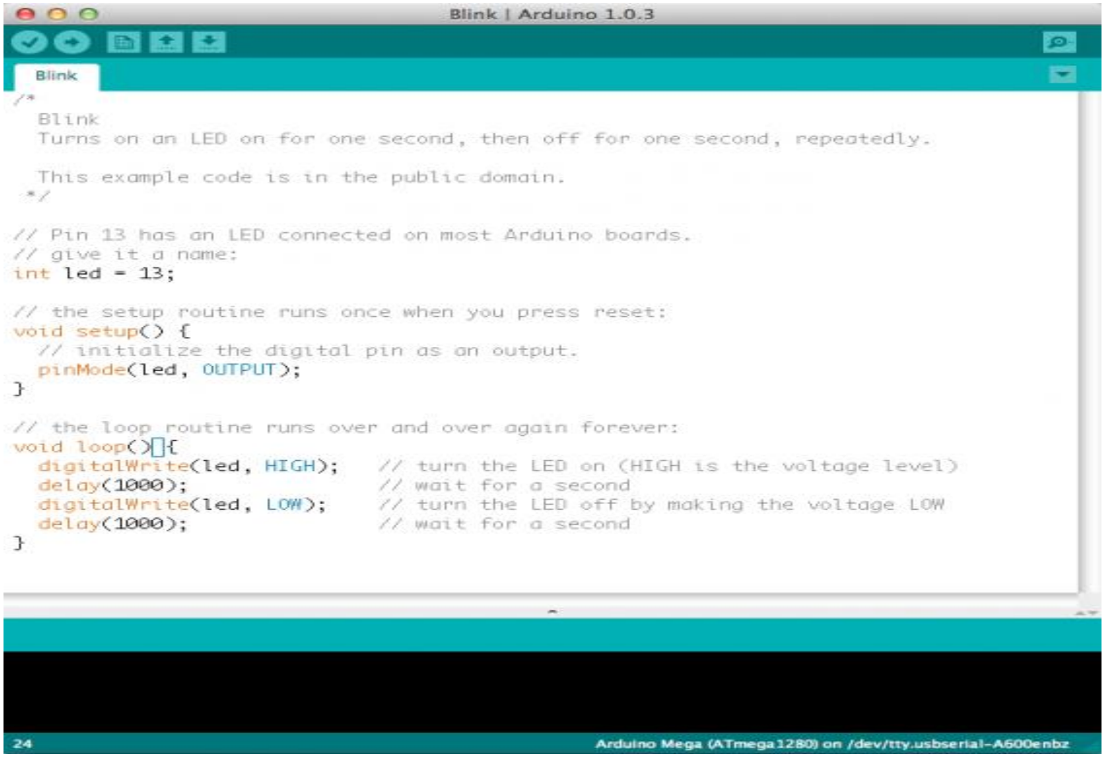


Figure 3.1.1: Arduino IDE.

(II) MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with

matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.
The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other

The MATLAB system consists of five main parts:

1. The MATLAB language.

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

2. The MATLAB working environment.

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

3.1.2 External interface requirement

3.1.2.2 Hardware Interfaces

I) Arduino Due (Arm Cortex M3)

The Arduino Due is the first Arduino board based on a 32-bit ARM core microcontroller. With 54 digital input/output pins, 12 analog inputs, it is the perfect board for powerful larger scale projects.

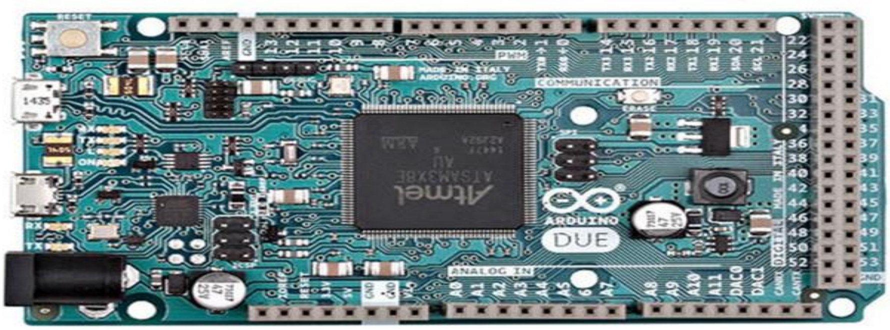


Figure 3.1.2.2: Arduino Due

Overview

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU (Fig. 3.1). It is the first Arduino board based on a 32-bit ARM core microcontroller. It has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), a 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button.

Unlike most Arduino boards, the Arduino Due board runs at 3.3V. The maximum voltage that the I/O pins can tolerate is 3.3V. Applying voltages higher than 3.3V to any I/O pin could damage the board.

The board contains everything needed to support the microcontroller; simply connect it to a computer with a micro-USB cable or power it with a AC-to-DC adapter or battery to get started. The Due is compatible with all Arduino shields that work at 3.3V and are compliant with the 1.0 Arduino pinout.

The Due follows the 1.0 pinout:

- TWI: SDA and SCL pins that are near to the AREF pin.
- IOREF: allows an attached shield with the proper configuration to adapt to the voltage provided by the board. This enables shield compatibility with a 3.3V board like the Due and AVR-based boards which operate at 5V.
- An unconnected pin, reserved for future use.

Specifications	Details
Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

Table3.1.2.2: Technical Specifications and Details-Arduino Due

II) LSA08 line sensor

Figure 3.1.2.2: LSA08 Line sensor

LSA08 (Fig. 3.2) consist of 8 sensors pair. LSA08 is typically used for embedded system or robots for line following task. The specially selected wavelength of super bright green LED as the sensor's transmitter enables LSA08 to operate on various different colour surfaces. LSA08 is capable to operate on surface with colour of Red, Green, Blue, White, Black, Gray and possibly other colours with distinct brightness different. LSA08 has several different output modes, for the convenience of use for any system. Namely, the digital output port (8 parallel output line), the serial communication port (UART) and the analog output port.

Features:

- 8 sensor pairs spaced 16mm.
- 12V input power
- On board Mode and Select button for instant configuration of LSA08
- 3 Different output modes (digital output port, UART output port, analog output port)
- LCD display unit showing 8 sensors analog value with bar chart and line position.
- Simple Auto-Calibration function to the line following surface.
- Junction Pulse (JPULSE) for detecting junction crossing and junction counting
- Power polarity protection
- Low current consumption (typically 26mA)

III) Motor Drivers

Figure 3.1.2.2: Motor Drivers

Introduction

TB6600 Arduino Stepper Motor Driver is an easy-to-use professional stepper motor driver, which could control a two-phase stepping motor. It is compatible with Arduino and other microcontrollers that can output a 5V digital pulse signal. TB6600 Arduino-stepper motor driver has a wide range power input, 9~42VDC power supply. And it is able to output 4A peak current, which is enough for the most of stepper motors.
The stepper driver supports speed and direction control. You can set its micro step and output current with 6 DIP switches. There are 7 kinds of micro steps (1, 2 / A, 2 / B, 4, 8, 16, 32) and 8 kinds of current control (0.5A, 1A, 1.5A, 2A, 2.5A, 2.8A, 3.0A, 3.5A) in all. And all signal

terminals adopt high-speed optocoupler isolation, enhancing its anti-high-frequency interference ability.
As a professional device, it is able to drive 57, 42-type two-phase, four-phase, hybrid stepper motor.

Features

- Support 8 kinds of current control
- Support 7 kinds of micro steps adjustable
- The interface adopts high-speed optocoupler isolation
- Automatic semi-flow to reduce heat
- Large area heat sink
- Anti-high-frequency interference ability
- Input anti-reverse protection
- Overheat, over current and short circuit protection

Specification

Specification	Details
Input Current:	0~5A
Output Current:	0.5~4.0A
Control Signal	3.3~24V
Power (MAX):	160W
Micro Step:	1, 2/A, 2/B, 4, 8, 16, 32
Temperature:	-10～45℃
Humidity:	No Condensation
Weight:	0.2 kg
Dimension:	96 * 71 * 37 mm

Table 3.1.2.2: Technical Specifications and Details- Motor Driver

IV) Stepper Motors

Figure 3.1.2.2 Stepper Motor

Stepper Motor Basics

A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotations. Each rotation of a stepper motor is divided into a set number of steps, sometimes as many as 200 steps. The stepper motor must be sent a separate pulse for each step. The stepper motor can only receive one pulse and take one step at a time and each step must be the same length. Since each pulse results in the motor rotating a precise angle — typically 1.8 degrees — you can precisely control the position of the stepper motor without any feedback mechanism.

As the digital pulses from the controller increase in frequency, the stepping movement converts into a continuous rotation with the velocity of the rotation directly proportional to the frequency of the control pulses. Stepper motors are widely used because of their low cost, high reliability, and high torque at low speeds. Their rugged construction enables you to use stepper motors in a wide environmental range.

Advantages of Using Stepper Motors

- A wide range of rotational speeds can be utilized since the speed of a step motor is proportional to the frequency of the input pulses from your controller.

- Precise open-loop positional control is possible with a stepper motor without any feedback mechanism.
- Very low speed rotation is possible with a load that is coupled directly to the shaft of the stepper motor.
- A stepper motor is quite reliable because there are no contact brushes. Generally, the life of a stepper motor is determined by the life of the stepper motor bearing.
- A stepper motor is very good at starting, stopping, and reversing direction.
- A stepper motor provides precise positioning and repeatability of movement.
- An energized stepper motor maintains full torque at standstill position.

Types of Stepper Motors

There are three kinds of step motors: permanent magnet, hybrid, and variable reluctance. Hybrid step motors offer the most versatility and combine the best characteristics of variable reluctance and permanent magnet stepper motors. Hybrid stepper motors are constructed with multi-toothed stator poles and a permanent magnet rotor. A standard hybrid stepper motor has 200 rotor teeth and rotates 1.8 degrees per step. Hybrid stepper motors provide high static and dynamic torque and they run at very high step rates. Applications for hybrid stepper motors include computer disk drives and cd players. Hybrid stepper motors are also widely used in industrial and scientific applications. Hybrid step motors are used in robotics, motion control, automated wire cutting, and even in high-speed fluid dispensers.

Step Modes

Stepper motor "step modes" include full step, half step, and micro step. The type of step is dependent on the stepper motor driver controlling the stepper motor. Many stepper motor controllers are multi-step capable (usually adjusted by switch setting).

i) Full Step

Standard hybrid stepping motors have 200 full steps per revolution. If you divide the 200 steps into the 360 degrees of rotation you get 200 1.8-degree steps. Normally this is achieved by energizing both windings while alternately reversing the current, meaning one pulse from the driver is equal to one full step on the step motor.

ii) Half Step

Half Step means that the stepping motor is rotating at 400 steps per revolution (0.9-degree steps x 400 = 360 degrees). First one winding is energized and then two windings are alternately energized. This will cause the rotor of the stepping motor to move at half the distance (0.9 degrees). In half-step mode, a typical stepper motor provides about 30% less torque, but it provides a smoother motion than it would in full-step mode.

iii) Micro-step

Micro-stepping is a relatively new stepping motor system. Micro-stepping energizes the stepper motor winding in a manner that further subdivides the number of positions between poles. Some micro-stepping controllers are capable of dividing a full step (1.8 deg) into 256 micro-steps. This would result in 51,200 steps in one revolution (.007 deg/step). Micro-stepping is usually applied to applications that require accurate positioning and smoother motion over a broad range of speeds. As in the half-step mode, micro-stepping reduces torque by about 30% compared to full-step mode.

3.2 Bluetooth modules

Bluetooth is a **standardized protocol** for sending and receiving data via a 2.4GHz wireless link. It’s a secure protocol, and it’s perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices.

Figure 3.2 Bluetooth

These days it feels like *everything* is wireless, and Bluetooth is a big part of that wireless revolution. You’ll find Bluetooth embedded into a great variety of consumer products, like headsets, video game controllers, or (of course) livestock trackers.

In our world of embedded electronics hackery, Bluetooth serves as an excellent protocol for wirelessly transmitting relatively small amounts of data over a short range (<100m). It's perfectly suited as a wireless replacement for serial communication interfaces. Or you can use it to create a DIY HID Computer Keyboard. Or, with the right module, it can be used to build a homebrew, wireless MP3-playing speaker.
This tutorial aims to provide a quick overview of the Bluetooth protocol. We'll examine the specifications and profiles that form its foundation, and we'll go over how Bluetooth compares to other wireless protocols.

II) IR sensors

An Infrared light emitting diode (IR LED) is a special purpose LED emitting infrared rays ranging 700 nm to 1 mm wavelength. Different IR LEDs may produce infrared light of differing wavelengths, just like different LEDs produce light of different colors. IR LEDs are usually made of gallium arsenide or aluminum gallium arsenide. In complement with IR receivers, these are commonly used as sensors.
The appearance of IR LED is same as a common LED. Since the human eye cannot see the infrared radiations, it is not possible for a person to identify if an IR LED is working. A camera on a cell phone camera solves this problem. The IR rays from the IR LED in the circuit are shown in the camera.

Pin Diagram of IR LED

An IR LED is a type of diode or simple semiconductor. Electric current is allowed to flow in only one direction in diodes. As the current flows, electrons fall from one part of the diode into holes on another part. In order to fall into these holes, the electrons must shed energy in the form of photons, which produce light.

It is necessary to modulate the emission from IR diode to use it in electronic application to prevent spurious triggering. Modulation makes the signal from IR LED stand out above the noise. Infrared diodes have a package that is opaque to visible light but transparent to infrared. The massive use of IR LEDs in remote controls and safety alarm systems has drastically reduced the pricing of IR diodes in the market.

Principle of Working

An IR sensor consists of two parts, the emitter circuit and the receiver circuit as shown in figure 3.8. This is collectively known as a photo-coupler or an optocoupler. The emitter is an IR LED and the detector is an IR photodiode. The IR photodiode is sensitive to the IR light emitted by an IR LED. The photo-diode's resistance and output voltage change in proportion to the IR light received. This is the underlying working principle of the IR sensor.

The type of incidence can be direct incidence or indirect incidence. In direct incidence, the IR LED is placed in front of a photodiode with no obstacle in between. In indirect incidence, both the diodes are placed side by side with an opaque object in front of the sensor. The light from the IR LED hits the opaque surface and reflects back to the photodiode.

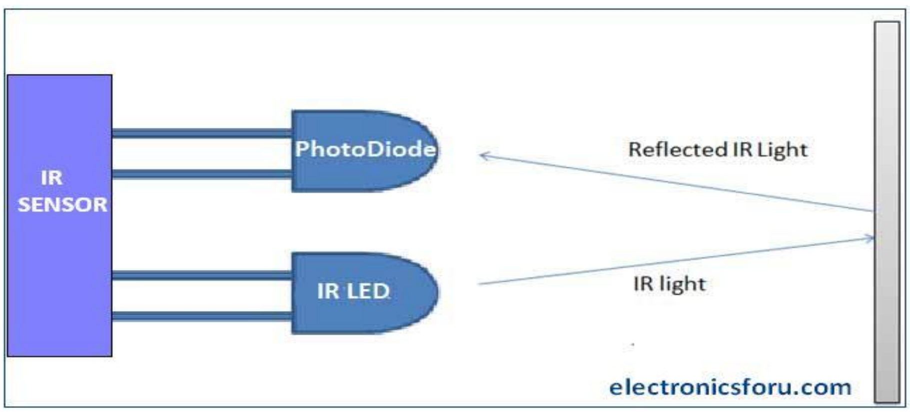


Figure 3.2:IR Sensor

Step by step details to build an IR sensor are available at: DIY- IR Sensor IR sensors find a wide variety of applications in various fields. Let's take a look at few of them.

III) I2C Communication

The Inter-Integrated Circuit (I^2C) Protocol is a protocol intended to allow multiple "slave" digital integrated circuits ("chips") to communicate with one or more "master" chips. Like the Serial

Peripheral Interface (SPI), it is only intended for short distance communications within a single device. Like Asynchronous Serial Interfaces (such as RS-232 or UARTs), it only requires two signal wires to exchange information.

Because serial ports are asynchronous (no clock data is transmitted), devices using them must agree ahead of time on a data rate. The two devices must also have clocks that are close to the same rate, and will remain so–excessive differences between clock rates on either end will cause garbled data.

Asynchronous serial ports require hardware overhead–the UART at either end is relatively complex and difficult to accurately implement in software if necessary. At least one start and stop bit is a part of each frame of data, meaning that 10 bits of transmission time are required for each 8 bits of data sent, which eats into the data rate.

Another core fault in asynchronous serial ports is that they are inherently suited to communications between two, and only two, devices. While it is possible to connect multiple devices to a single serial port, bus contention (where two devices attempt to drive the same line at the same time) is always an issue and must be dealt with carefully to prevent damage to the devices in question, usually through external hardware.

Finally, data rate is an issue. While there is no theoretical limit to asynchronous serial communications, most UART devices only support a certain set of fixed baud rates, and the highest of these is usually around 230400 bits per second.

3.3 System Features

3.3.1 Feature 1: System will be able to line trace and make movements and displace from one place to another.

3.3.2 Feature 2: System will be able to AI to make decisions i.e. Zone shifting.

3.4 Other Non- functional requirements

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions.

3.5 Performance requirements

1.Quick and real-time responses

The robot should be able to make the responses to the stimuli and accordingly perform the action. The robot will make real time decisions to move to the different zones based on the points scored.

2.Fast and serial communication

The Universal Asynchronous Receiver/Transmitter (UART) controller is the key component of the serial communications subsystem of a computer. UART is also a common integrated feature in most microcontrollers. The UART takes bytes of data and transmits the individual bits in a sequential fashion. High speed serial communication should be there in order to ensure that robot throws the shuttlecock with accuracy and speed.

3.6 Safety Requirements

1.Emergency Switch

Emergency stop buttons will there on the robots. The robots will be designed and built so as to pose danger to no one, including the team, the opposing team, the people in the surroundings, and the venue.

2.Reset button on Controller

Use a small, unfolded paper-clip or something similar to click the button (the button is inside a tiny hole). After resetting the controller, re-pair the controller with connecting the USB cable to the controller and the system and pressing the start button.

3.7 Security Requirements

1. Bluetooth Master Recognition

One of the complexities often associated with wireless technology is the process of connecting wireless devices. Users have become accustomed to the process of connecting wired devices by plugging one end of a cable into one device and the other end into the complementary device. Hence a master recognition is required, Bluetooth technology uses the principles of device "inquiry" and "inquiry scan." Scanning devices listen in on known frequencies for devices that are actively inquiring. When an inquiry is received, the scanning device sends a response with the information needed for the inquiring device to determine and display the nature of the device that has recognized its signal.

3.8 Risk Assessment

3.7.1 Risk Management w.r.t. NP Hard analysis:

The project manager working with the project team and project sponsors will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risks will be identified as early as possible in the project so as to minimize their impact.

3.7.2 Risk Identification

Risks are identified and their parameters are explained in the Table 3.7.1.

ID	Risk Description	Probability	Impact	Schedule	Quality	Overall
1)	The battery of the robot might drain	Moderate	High	Efficient charging Stations	Low	High
2)	Hardware Components might get damaged due to over heating	High	high	Time instance	Moderate	Moderate
3)	Sensors might not work for tracing the line	Low	High	Time instance	Moderate	Moderate

Table 3.7.1 Risk Management

3.7.3 Risk Analysis

All risks identified will be assessed to identify the range of possible project outcomes. Qualification will be used to determine which risks are the top risks to pursue and respond to and which risks can be ignored. Qualitative Risk Analysis The probability and impact occurrence for each identified risk will be assessed by the project manager, with input from the project team using the following approach:

Impact:

• High Risk: That has the potential to greatly impact project cost, project schedule or performance

• Medium Risk: That has the potential to slightly impact project cost, project schedule or performance

• Low Risk: That has relatively little impact on cost, schedule or performance

Risks that fall within the RED and YELLOW zones will have risk response planning which may include both risk mitigation and a risk contingency plan.

Quantitative Risk Analysis of risk events that have been prioritized using the qualitative risk analysis process and their affect on project activities will be estimated, a numerical rating applied to each risk based on this analysis, and then documented in this section of the risk management plan.

Chapter 4
Modelling

1 UML Diagram

4.1.1 Used Case Diagram for automatic and manual robot

The robotic system use case involves primary actors: Providing input (who initiates the system)

4.1.2 State Diagram

A state diagram, also called a state machine diagram or state-chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML). For state diagram of the system, Refer 4.1.2

4.1.3 Class diagram

Class diagram is one of the dynamic model in UML diagrams, it represents all the attributes/classes and operations involved in the system as well as shows relationship between those classes using the relationship association. Class diagram shows static view of a software, where different class which are present in the system are described.

4.1.4 Activity diagram

Represents dynamic view of the system. It contains object and activity. Also it follows fork join model.

4.1.1 Use Case Diagram

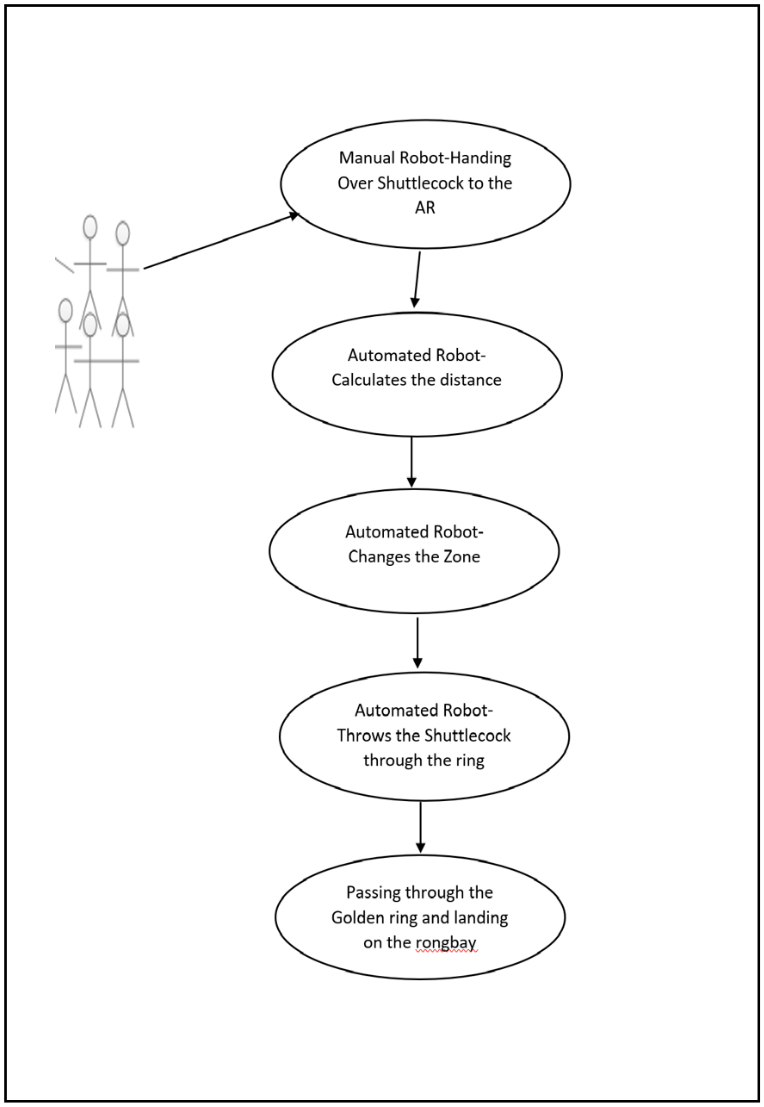


Figure 4.1.1 Used Case Diagram for automatic and manual robot

In figure 4.1.1 there are mainly two actors interacting with the robotic system

Primary Actor: User providing input

Secondary Actor: User to controlling the AR

Actions and interactions: Hand-over of shuttle-cock, throwing of shuttle-cock, changing of various zones.

4.1.2 State Diagram

State Diagram consist of two Robots, Automated robot and manual Robot The state diagram represents the separate independent states in the system and shows the interaction and data flow between the various states, as shown in figure 4.1.2. The arrow represents the direction of flow of data and text on arrow denotes action or task taking place during transition.

All the actions have their own implications and sand functionalities. Using manual robot line tracing can be done and automated robot throws the shuttle cock. After passing through the ring the goal state is reached.

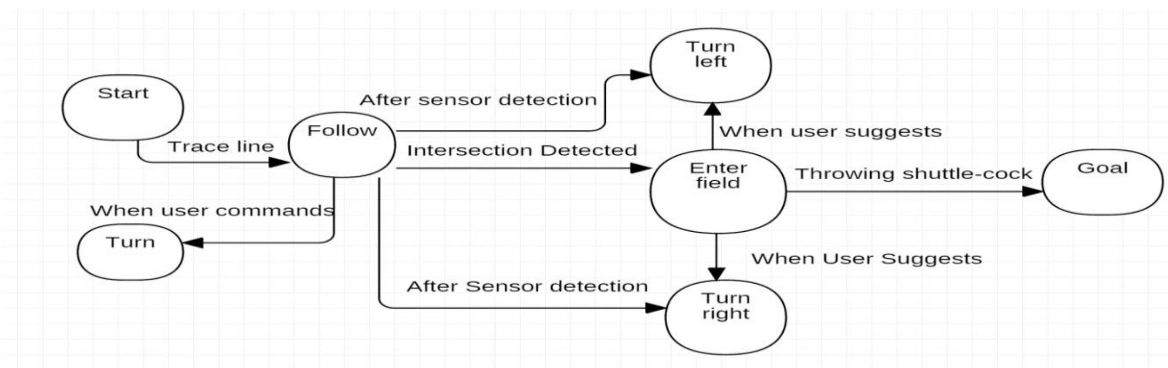


Figure 4.1.2 State Diagram

4.1.3 Class Diagram

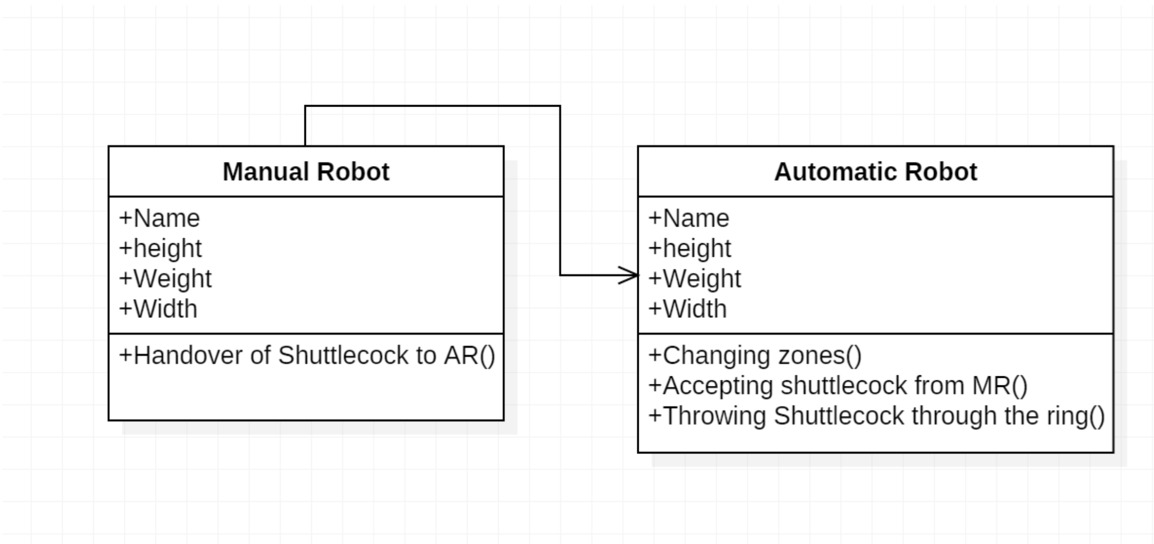


Figure 4.1.3 Class diagram

Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them, refer figure 4.1.3.

- **Name:** There are two classes Manual Robot class and Automatic Robot class
- **Attributes:** The second row in a class shape. Each attribute of the class is displayed on a separate line.
- **Methods/Operations:** The third row in a class shape. Also known as operations, methods are displayed in list format with each operation on its own line.

4.1.4 ACTIVITY DIAGRAM

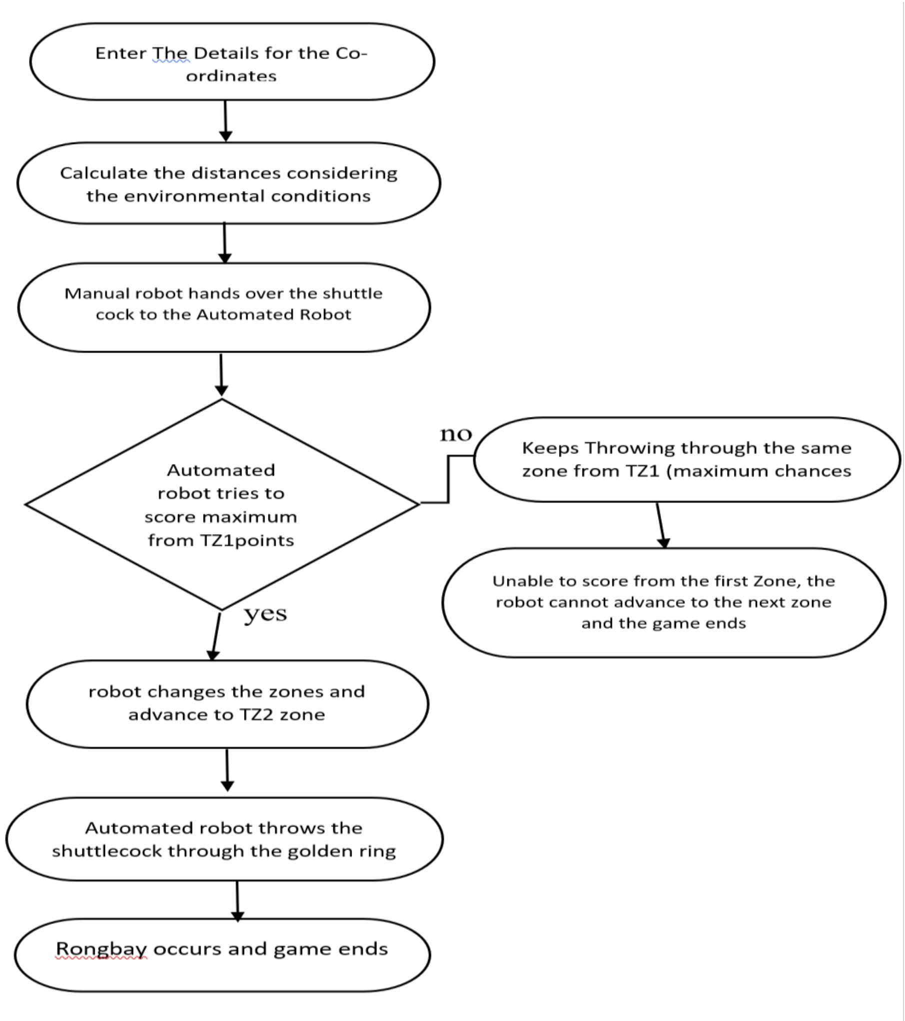


Figure 4.1.4 Activity Diagram for automatic and manual robot

This activity diagram figure 4.1.4 is used to model the activities which are nothing but robot moving from different zones and scoring points. The diagram has more impact on understanding rather than on implementation details.

- Activities: Entering zones, scoring points, Throwing the shuttle-cock, calculating the distances
- Conditions: The robot can move to a different zone only after scoring a point from the current zone. i.e. it can proceed from TR2 only after scoring from TR1
- Constraints: The robot cannot weigh more than 25 kgs.

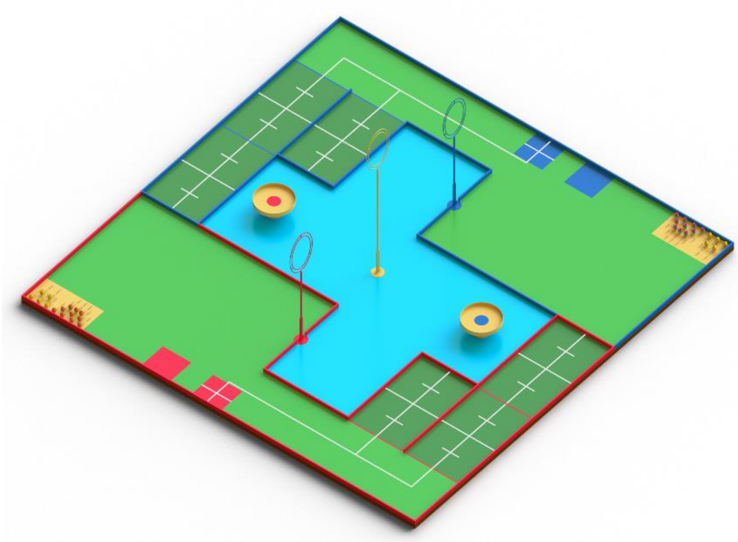


Figure 4.4 Game Field: Isometric View

- A game between two teams takes place within 3 minutes. Each team has two robot of:
- One (1) manual robot and one (1) automatic robot or
- Two (2) automatic robots.
- Only one (1) automatic robot is allowed to throw Shuttlecock.

- Game filed is divided into 3 areas, including fields for the 2 teams and NC area (See figure 5.4).

- A team field consists of: Start Zone, Loading Zone, Throwing Zone, Manual Robot Area, and Automatic Robot Area.

- NC Area is placed with Ring Trees, Normal Rings, Golden Ring and Golden Cup.

- When a game starts, Manual Robot will pick Normal Shuttlecocks and handle it to Automatic Robot.
- After receiving the Normal Shuttlecock, Automatic Robot will move into TZ1, TZ2 and throw the shuttlecock at the Normal Ring. If the shuttlecock goes through the ring successfully, points will be given.

- Manual Robot can go and pick up the Golden Shuttlecock only after at least one Normal Shuttlecock thrown from each TZ1 and TZ2 went through the Normal Ring successfully.

- After receiving Golden Shuttlecock from Manual Robot, Automatic Robot can move to TZ3 and throw the Golden Shuttlecock at the Golden Ring. If the Golden Shuttlecock goes through the Golden Ring successfully, points will be given.

- When Golden Shuttlecock is thrown through the Golden Ring and then is landed on the Golden Cup, that team will gain the victory and the match will be finished. This victory is called ***"Rongbay"*** ("Flying Dragon").

- If neither team reaches ***"Rong bay"***, and the game time of three (3) minutes passes, the game shall end. The winner will be decided by who has the higher score at the said end of the game.

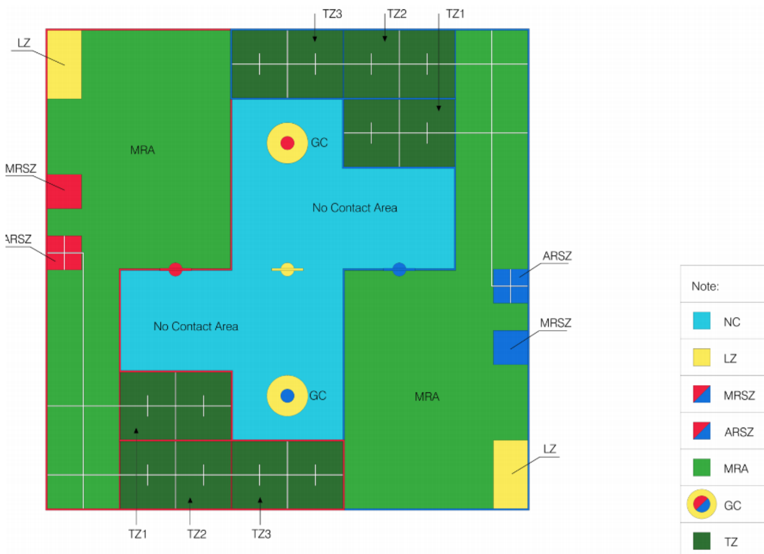


Figure 4.5 Zones and areas on game field

Terms	Definitions
Manual Robot	The robot which is operated by operator via wireless or cable connection. Abbreviation: MR.
Automatic Robot	The robot which is able to work independently without any help from an operator. Abbreviation: AR
Manual Robot Start Zone	An area, from where the manual robot starts the game. Abbreviation: MRSZ.
Automatic Robot Start Zone	An area, from where the Automatic Robot starts the game. Abbreviation: ARSZ.
No Contact Area	An area which robots cannot come in contact with.

	Robots are able to enter the space above. Abbreviation: NC.
Throwing Area	An area from which Automatic Robot throws Shuttlecock. Throwing Area consists of three zones: - The first throwing zone: Abbreviation: TZ1. - The second throwing zone: Abbreviation: TZ2. - The third throwing zone: Abbreviation: TZ3.
Loading Zone	Areas where teams allocate Shuttlecock or Shuttlecock Rack before the game begins. Abbreviation: LZ.

Table 4.1: Play zones and Descriptions

Chapter 5

Coding

5.1 Algorithm

PID -Proportional Integral Derivative. As the name suggests, these terms describe three basic mathematical functions applied to the error (error = SetVal - SensorVal, where SetVal is the target value and SensorVal is the present input value obtained from the sensor). Main task of the PID controller is to minimize the error of whatever we are controlling. It takes in input, calculates the deviation from the intended behavior and accordingly adjusts the output so that deviation from the intended behavior is minimized and greater accuracy obtained.

Line following seems to be accurate when carried out at lower speeds. As we start increasing the speed of the robot, it wobbles a lot and is often found getting off track. Hence some kind of control on the robot is required that would enable us to make it follow the line efficiently at higher speeds. This is where PID controller shines.

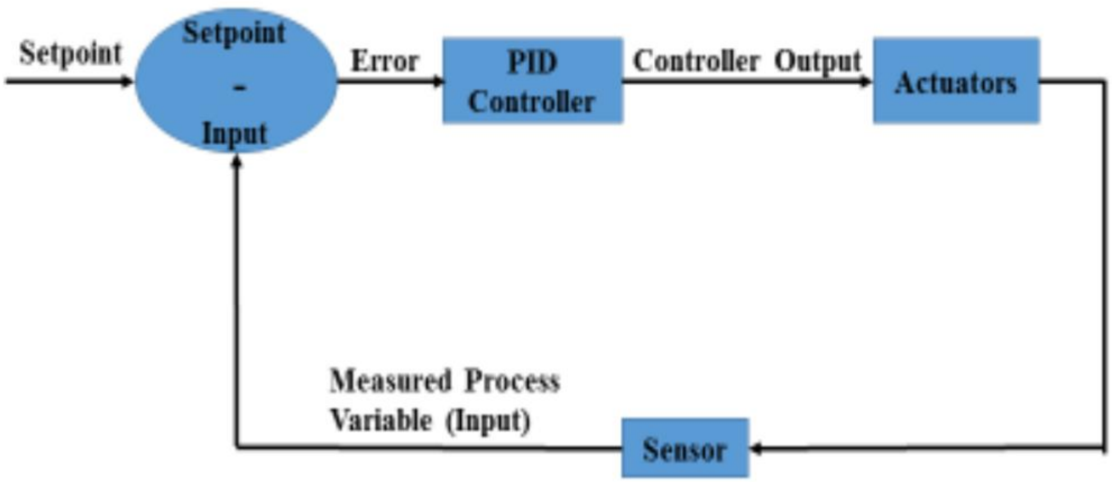


Figure 5.1 PID controller.

In order to implement line following one can basically start with just three sensors which are so spaced on the robot that-

1. If the center sensor detects the line the robot steers forward
2. If the left sensor detects the line the robot steers right
3. If the right sensor detects the line the robot steers left.

This algorithm would make the robot follow the line, however, we would need to compromise with its speed to follow the line efficiently.

There will be other possible combinations such as 00110 and 00011 that can provide us data on how far to the right is the robot from the center of the line (same follows for left). Further to implement better line following we need to keep track of how long is the robot not centered on the line and how fast does it change its position from the Centre. This is exactly what we can achieve using “PID” control. The data obtained from the array of sensors would then be put into utmost use and line following process would be much smoother, faster and efficient at greater speeds.

PID is all about improving our control on the robot.

The idea behind PID control is that we set a value that we want maintained, either speed of a motor or reading from a sensor. We then take the present readings as input and compare them to the setpoint. From this an error value can be calculated, i.e., (error = setpoint - actual reading). This error value is then used to calculate how much to alter the output by to make the actual reading closer to the setpoint.

How to Implement PID

Terminology:

The basic terminology that one would require to understand PID are:

- Error - The error is the amount at which a device isn’t doing something right. For example, suppose the robot is located at x=5 but it should be at x=7, then the error is 2.
- Proportional (P) - The proportional term is directly proportional to the error at present.

- Integral (I) - The integral term depends on the cumulative error made over a period of time (t).
- Derivative (D) - The derivative term depends rate of change of error.
- Constant (factor)- Each term (P, I, D) will need to be tweaked in the code. Hence, they are included in the code by multiplying with respective constants.
    - P-Factor (Kp) - A constant value used to increase or decrease the impact of Proportional
    - I-Factor (Ki) - A constant value used to increase or decrease the impact of Integral
    - D-Factor (Kd) - A constant value used to increase or decrease the impact of Derivative

Error measurement: In order to measure the error from the set position, i.e. the center we can use the weighted values method. Suppose we are using a 5-sensor array to take the position input of the robot. The input obtained can be weighted depending on the possible combinations of input. The weight values assigned would be such that the error in position is defined both exactly and relatively.

The full range of weighted values is shown below. We assign a numerical value to each one.

PID formula:

So what do we do with the error value to calculate how much the output be altered by? We would need to simply add the error value to the output to adjust the robot's motion. And this would work, and is known as proportional control (the P in PID). It is often necessary to scale the error value before adding it to the output by using the constant(Kp).

Proportional:

Difference = (Target Position) - (Measured Position)
Proportional = Kp*(Difference)

This approach would work, but it is found that if we want a quick response time, by using a large constant, or if the error is very large, the output may overshoot from the set value. Hence the change in output may turn out to be unpredictable and oscillating. In order to control this, derivative expression comes to limelight.

Derivative:

Derivative provides us the rate of change of error. This would help us know how quickly does the error change from time to time and accordingly we can set the output.

Rate of Change=((Difference) – (Previous Difference))/time interval
Derivative= Kd *(Rate of Change)

The time interval can be obtained by using the timer of microcontroller.

The integral improves steady state performance, i.e. when the output is steady how far away is it from the setpoint. By adding together all previous errors it is possible to monitor if there are accumulating errors. For example- if the position is slightly to the right all the time, the error will always be positive so the sum of the errors will get bigger, the inverse is true if position is always to the left. This can be monitored and used to further improve the accuracy of line following.

Integral:

Integral=Integral+Difference
Integral = Ki*(Integral)

Therefore, Control value used to adjust the robot’s motion=

(Proportional) + (Integral) + (Derivative)

5.2 Software Used

Arduino Software (IDE)

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension no. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor as shown in figure 5.2.

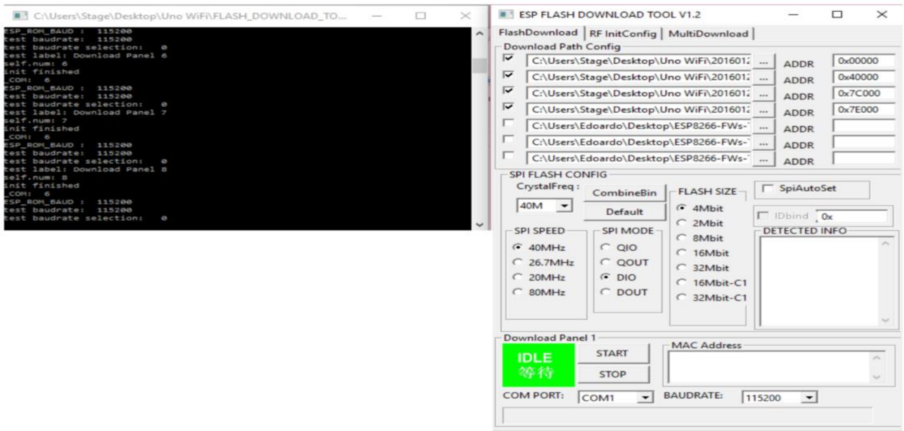


Figure 5.2: Arduino Software IDE

5.3 Hardware Specification

Serial Monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial. Begin in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor.

Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Arduino Software (IDE) includes the built-in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

5.5 Components and Tools Used

1. Arduino Due (Arm Cortex M3)
2. LSA08 line sensor
3. Motor Drivers
4. MATLAB
5. Arduino Uno
6.Ardunio IDE

Description of the components in the system

The control designs generally consist of microcontrollers, actuators, and sensors. The Arduino UNO and TEC chosen as microcontroller and actuators respectively.

Arduino UNO:

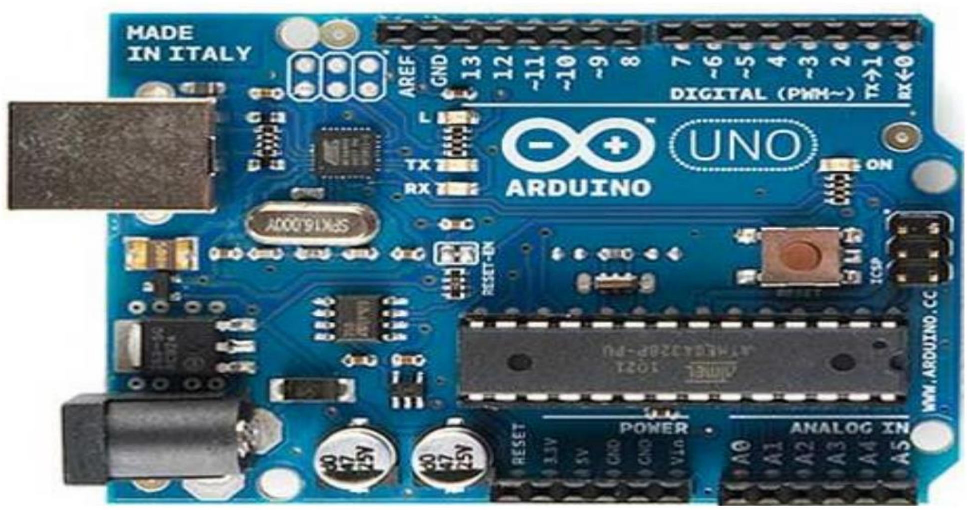


Figure 5.5 Arduino UNO

An Arduino UNO is a microcontroller which operates by reading the sensor values, making logical decisions, and sending the necessary signals to the actuators. A brief description of Arduino UNO shown in Fig 2.2 is as follows:

1. It has a USB power plug and a separate power plug. The separate power plug takes the power supply through the external voltage source.
2. It has five analog input pins that measure the signals from the sensors.

3. It has a USB power plug and a separate power plug. The separate power plug takes the power supply through the external voltage source.
4. It has twelve digital pins for digital input/output (2-13), among them five pins (3, 6, 9, 10 and 11) are PWM pins.
5. Digital pins 0 and 1 are labeled RX and TX respectively, which are serial in and serial out pins. There are three ground pins, one input voltage pin, one 5 V pin, one 3.3 V pin, one reset push-button, and one Analog Reference (AREF) pin.
6. Atmega328 is a microprocessor used with an In Circuit Serial Programmer (ICSP). In our experimental setup, the code written in a C programming language (Arduino software) is uploaded to the Arduino UNO instead of using the PID library.
7. In our project, Arduino UNO is basically performing the following three tasks:

   - Reading the analog signals (values) from the temperature sensor.
   - After comparing the received signal with the desired signal, sending digital output as PWM signals to the actuators.
   - Printing the data as text-files, the process related to data acquisition.

Arduino-based PID controllers

A discussion of some of the exciting works with the Arduino-based PID controllers was made . The Arduino is an open source device which is cheap (price ranges from approximately $24 for an Arduino-UNO to almost $44 for the Arduino-Mega) and is user friendly, making it popular among hobbyists. Moreover, it can be programmed using the simple programming language “C/C++,” and can be interfaced easily with other advanced computing environments like MATLAB and LABVIEW. The range of applications of the Arduino-based PID controller varies from a simple temperature controller to a robot controller.

5.8 Coding Style Format

Stepper Mechanism:

```
#include <AltSoftSerial.h>
#include <Stepper.h>
int stepsPerRevolution = 1600;
AltSoftSerial Aserial;
Stepper myStepper(stepsPerRevolution,2,3);
```

```
int val,pval;
void setup() {

  Aserial.begin(9600);
  Serial.begin(9600);
  //delay(1000);
}
int Speed=0,prevmillis;
int adder=20;
int dir;
int switchMotor;
void loop() {

  if(Serial.available())
    val=Serial.read();

  if(val==1)
  dir=1;
  else if(val==2)
  dir=-1;

  if(val==3)
  switchMotor=1;
  else if(val==4)
  switchMotor=-1;

  //Serial.println(val);

  if(val<0){
      val=pval;
  }

  if(val>=30 && val<=255)
  {
    Speed=map(val,50,255,30,1000);


    myStepper.setSpeed(Speed);
    myStepper.step(dir*2);


  }
```

```
  else
     stepsPerRevolution=0;

  pval=val;
}
```

Chapter 6

Results and Discussion

This chapter includes result of each and every module involved in the system. The final output of the end product i.e the robot.

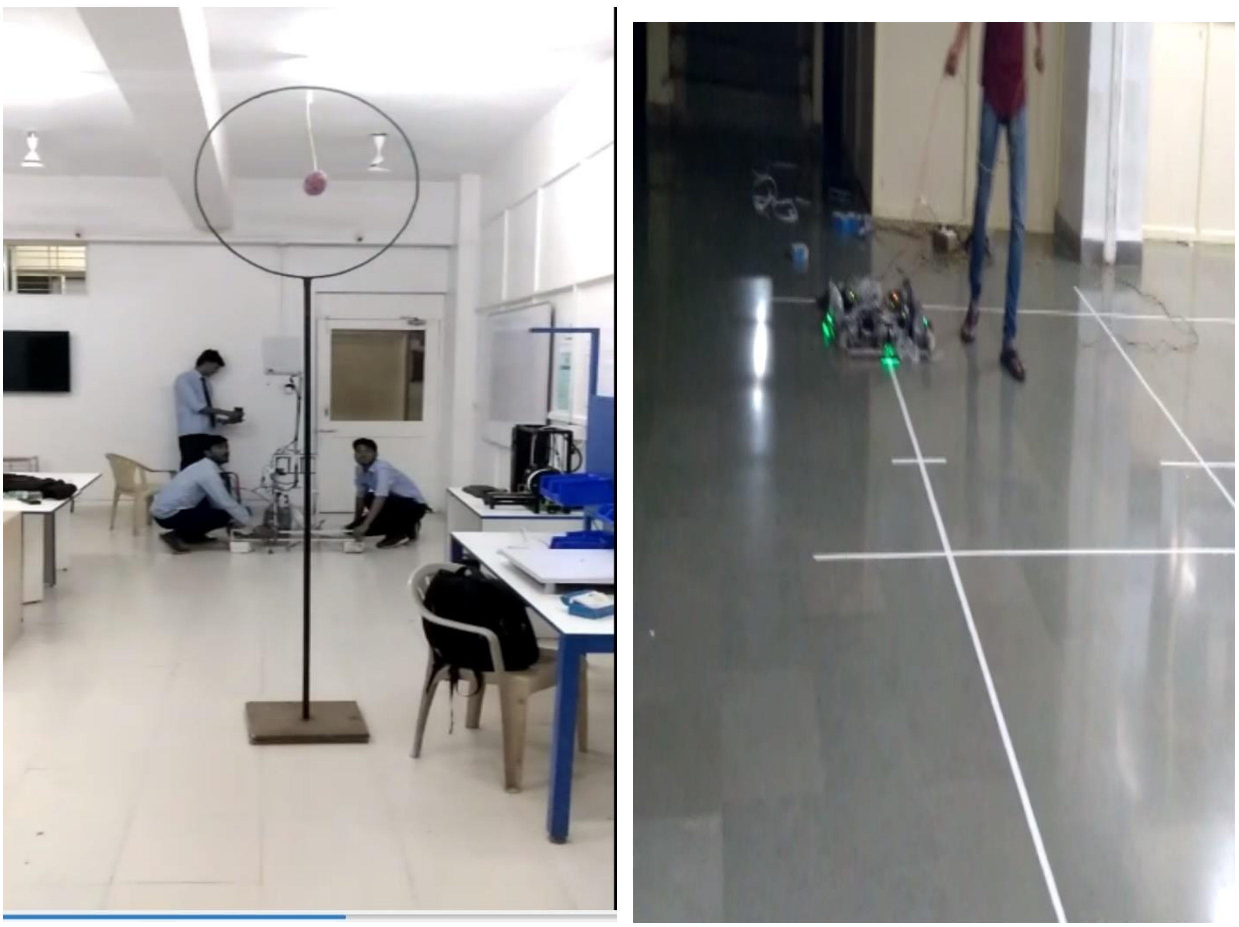


Figure 6.1 Automated Robot Throwing Shuttle cock and manual in Initial Stages of Development

Figure 6.1 Nirama University, Ahmedabad VS PCCOE Pune Super League Top-24 where PCCOE Team Automatons beat Nirma University .

Chapter 7

Testing

This chapter describes the testing methodologies and plans used for checking the performance and result of individual modules as well as the whole system. Test cases created and applied to separate components and integration of modules are also mentioned.

7.1 Test plan

7.1.1 Purpose

This document describes the plan for testing the robot:

• Identify the components of the project and test them in order to satisfy the requirements

. • List the recommended test strategies.

• Identify the required resources and provide an estimate of the test efforts.

7.1.2 Review requirement and Design Requirement

They are shown in the use case diagram which are used to refer while testing.As this project has many functions system is divided into many components.These individual components are tested as per their respective test cases.

7.1.3 Features to be tested Testing is preformed

On use case specification, functional requirement and nonfunctional requirement of all use cases. They include

• Manual Robot Testing

• PID Sensor Working

• Automated Robot testing

• Movements of robot in various directions

• Interfacing with controller

7.1.4 Approach Test strategies are basically the process to perform the recommended testing on the system.

1.Unit testing: Test the components of the system individually. The various components of the robotlike the sensors wheels etc. are individually tested. Expected output is compared with actual output.

2.System Testing: System should be tested against positive and negative inputs and then check if the system performs as expected.Warning messages must be displayed when needed. Error messages must be descriptive which would help it to be rectified as early as possible. **3.Performance Testing**: System performance can be tested by passing the shuttle-cock to the automated robot and throwing through the ring. The response time of passing the shuttle-cock and throwing action is measured and performance of system is calculated approximately.

4.Stress Testing: Stress testing considers conditions like low battery; connection strength of the Bluetooth modulate.

7.2 Test cases and Test Results

7.2.1 Manual Robot Testing

Test Case ID	001
Test Case Summary	To verify whether the manual robot is working successfully or not
Prerequisite	1. User knows how to operate the controller
Test Procedure	1.Connect the Controller to the manual robot 2.Power on the robot 3.Use the controller to make backward and forward movements
Expected Output	1.If robot makes movements in correct directions, it is successful. 2. If not, red light is displayed.
Actual Output	1. If robot makes movements in correct directions, the result is as expected 2. If not, nothing happens; the robot is still.
Status	Success
Test Environment	Floor with white stripes on it

Table 7.2.1 Manual Robot testing

Test Case ID	002
Test Case Summary	To verify whether PID Sensor is working
Prerequisite	1. User knows how to connect the sensor
Test Procedure	1.Connect the Controller to the manual robot 2.Power on the robot 3.Use the controller to make backward and forward movements
Expected Output	1.If robot makes movements in correct directions, it is successful. 2. If not, red light is displayed.
Actual Output	1. If robot makes movements in correct directions, the result is as expected 2. If not, nothing happens; the robot is still.
Status	Success
Test Environment	Floor with white stripes on it

Table 7.2.1 PID Sensor testing

Test Case ID	003
Test Case Summary	To verify whether Automated robot is working
Prerequisite	1. User knows how to control the robot
Test Procedure	1.Connect the Controller to the manual robot 2.Power on the robot 3.Pass the shuttle to the automated robot using manual robot
Expected Output	1.If robot throws the shuttle in correct directions, it is successful. 2. If not, it does not change the zones
Actual Output	1. If robot throws in correct directions, the result is as expected 2. If not, nothing happens; the robot is still.
Status	Success
Test Environment	Testing Arena was designed.

Table 7.2.1 Automated robot testing

Chapter 8

Configuration Management Plan

8.1 Installation of software:

1. When you plug an Arduino board into a Windows 10 computer, a driver should automatically be installed for it.

2. The driver configures the Arduino as a virtual COM port which can be seen in the Device Manager.

3. Open the Device Manager window by right-clicking the Windows 10 start button in the lower left of the screen and then selecting Device Manager on the menu that pops up.

4. In Device Manager, expand Ports (COM & LPT) and you should see a COM port which your Arduino will be as shown in the image below.

5. If you have other COM ports on the PC, then unplug the Arduino to see which COM port disappears from the device manager, then plug the Arduino back in and see which port appears – this will be the Arduino port, e.g. COM4

Installing Arduino Drivers

1. In the Device Manager, right-click the COM port that was identified as the Arduino (the Arduino must be plugged into the computer first). Select **Update Driver Software...** from the menu that pops up
2. After the driver has been installed, a dialog box will appear that shows that the driver has been successfully updated and will display the Arduino COM port number.
3. Finally back in the Device Manager, the COM port is now identified as an Arduino when the Arduino driver from the Arduino IDE folder is installed.

8.2 Un-installation:

On windows: Arduino IDE 1.6.5r5 and

previous:\Users\(username)\AppData\Roaming\Arduino15
ArduinoIDE1.6.6a and

later:\Users\(username)\AppData\Local\Arduino15

1. If you downloaded Admin Installation as called Windows Installer, you need to delete Arduino IDE from Control Panel.

2. If you downloaded non-Admin Installation, just delete the Arduino setup folder.

8.3 User help

1. The robot is mainly two components an automated component(which is the automated robot AR) and a manual component(which is the manual robot).
2. The manual robot is controlled by the player(human) using a PlayStation controller. to pass the shuttle cock-to the automated robot.
3. The automated robot is powered with AI and makes the decision on changing the zones based on the throwing of the shuttle-cock

CHAPTER 9

Conclusion

Conclusion is drawn on the basis of objectives. In this chapter at what extent conclusion are meeting the objective can be analyzed. In future scope, the future possible development in the project are discussed.

Considering the essential components, the proposed system of the robot will be helpful for line tracing purpose. There are many robots which can perform the task of throwing object using robotic arm. By this model we will be able to perform the task of throwing the shuttlecock by giving it a circular motion and performing the release of the shuttle. Along with this communication and understanding between the two robots will be achieved to execute the task successfully.

Obtaining accuracy and precision along with speed and minimum response time. This is achieved by this project.

Chapter 10

References

[1] DOF dynamic pitching robot that independently controls velocity, Angular velocity, and direction of a ball: Contact models and motion planning, Atari Mori; Jun Ueda; Tsukasa Ogasawara 2009 IEEE International Conference on Robotics and Automation.

[2] Accurate Object Throwing by an Industrial Robot Manipulator Wilhelm August, Steffen Waeldele, Bjoern Hein and Heinz WoernKarlsruhe Institute of Technology, Karlsruhe, Germanywilhelm.august@kit.edu,

[3] International Journal of Engineering Trends and Technology (IJETT) – Volume 21 Number 7 – March 2015 ISSN: 2231-5381.A Survey Paper on PID Control System Daya Sagar Sahu#1, Sunil Sharma*2 #M. Tech Scholar CS&E Dr. C.V. Raman university Bilaspur, * Asst. Prof. CS&E Dr. C.V. Raman University Bilaspur India.

[4] Research Paper STUDY OF TUNING OF PID CONTROLLER BY USING PARTICLE SWARM OPTIMIZATION Ankita Nayak1, Mahesh Singh2, Address for Correspondence 1M.E. Scholar, 2Sr. Ast. Professor, SSGI, SSTC India

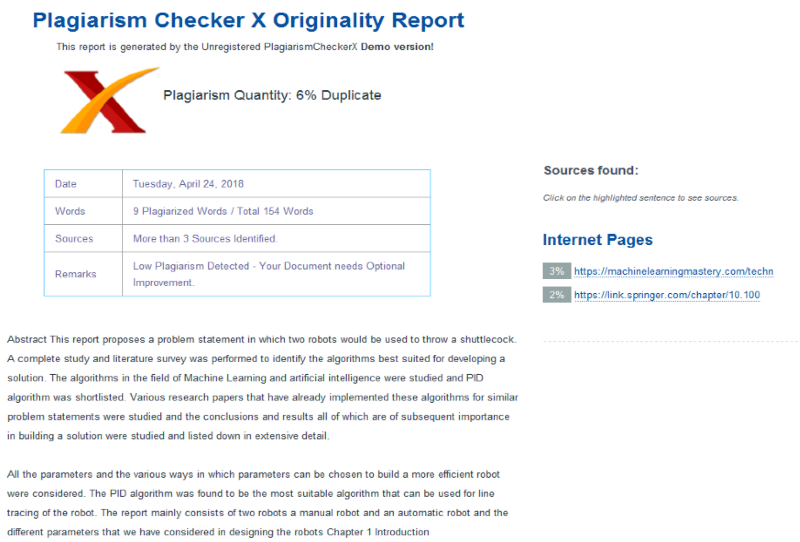
[5] Study Of tuning Of PID Controller By Using Particle Swarm Optimization, Ankita Nayak1 , Mahesh Singh M.E. Scholar, 2 Sr. Asstt. Professor, SSGI, SSTC India International Journal of Advanced Engineering Research and Studies

[6] PID Control System Analysis, Design, and Technology, Kiam Heong Ang, Gregory Chong, Student Member, IEEE, and Yun Li, Member, IEEEIEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 13, NO. 4, JULY 2005

[7] A Survey Paper on PID Control System, Daya Sagar Sahu, Sunil Sharma,International Journal of Engineering Trends and Technology (IJETT) – Volume 21 Number 7 – March 2015

CHAPTER 11

Plagiarism Check Report



Plagiarism Checker X Originality Report

This report is generated by the Unregistered PlagiarismCheckerX **Demo version!**

Plagiarism Quantity: 6% Duplicate

Date	Tuesday, April 24, 2018
Words	9 Plagiarized Words / Total 154 Words
Sources	More than 3 Sources Identified.
Remarks	Low Plagiarism Detected - Your Document needs Optional Improvement.

Sources found:

Click on the highlighted sentence to see sources.

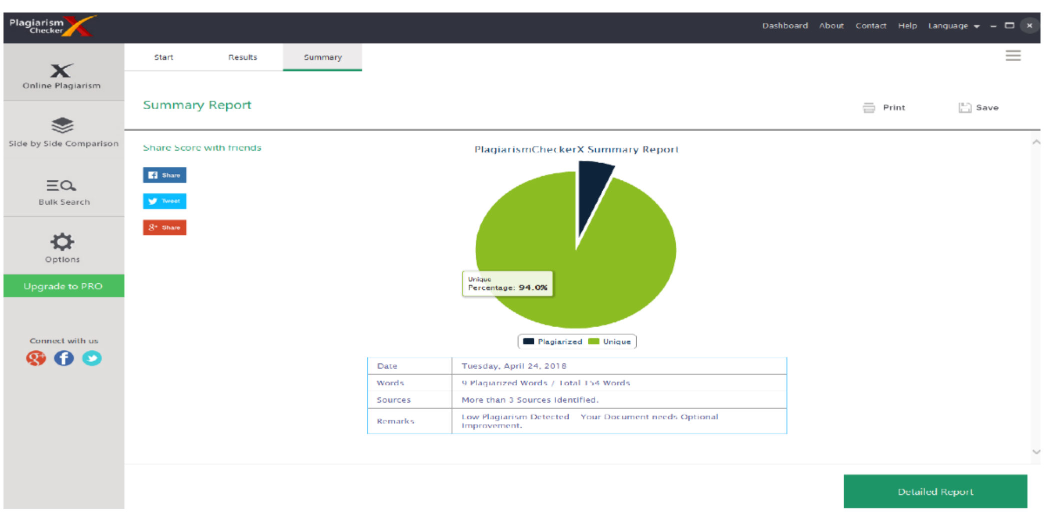
Internet Pages

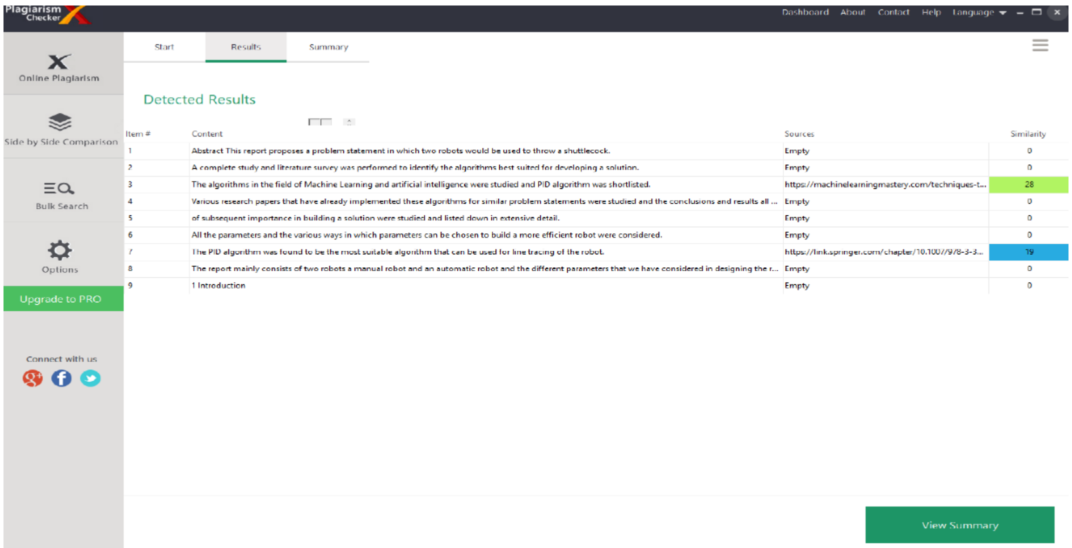
3% https://machinelearningmastery.com/techn

2% https://link.springer.com/chapter/10.100

Abstract This report proposes a problem statement in which two robots would be used to throw a shuttlecock. A complete study and literature survey was performed to identify the algorithms best suited for developing a solution. The algorithms in the field of Machine Learning and artificial intelligence were studied and PID algorithm was shortlisted. Various research papers that have already implemented these algorithms for similar problem statements were studied and the conclusions and results all of which are of subsequent importance in building a solution were studied and listed down in extensive detail.

All the parameters and the various ways in which parameters can be chosen to build a more efficient robot were considered. The PID algorithm was found to be the most suitable algorithm that can be used for line tracing of the robot. The report mainly consists of two robots a manual robot and an automatic robot and the different parameters that we have considered in designing the robots Chapter 1 Introduction




Plagiarism Checker
Dashboard About Contact Help Language
Start Results Summary
Online Plagiarism
Side by Side Comparison
Bulk Search
Options
Upgrade to PRO
Connect with us
Detected Results
Item # Content Sources Similarity
1 Abstract This report proposes a problem statement in which two robots would be used to throw a shuttlecock. Empty 0
2 A complete study and literature survey was performed to identify the algorithms best suited for developing a solution. Empty 0
3 The algorithms in the field of Machine Learning and artificial intelligence were studied and PID algorithm was shortlisted. https://machinelearningmastery.com/techniques-t... 28
4 Various research papers that have already implemented these algorithms for similar problem statements were studied and the conclusions and results all ... Empty 0
5 of subsequent importance in building a solution were studied and listed down in extensive detail. Empty 0
6 All the parameters and the various ways in which parameters can be chosen to build a more efficient robot were considered. Empty 0
7 The PID algorithm was found to be the most suitable algorithm that can be used for line tracing of the robot. https://link.springer.com/chapter/10.1007/978-3-3... 19
8 The report mainly consists of two robots a manual robot and an automatic robot and the different parameters that we have considered in designing the r... Empty 0
9 1 Introduction Empty 0
View Summary

Chapter 12
Term-II Project Laboratory
Assignments

- ASSIGNMENT 1

TITLE :- Review of design and necessary actions taking into consideration the feedback report of term II assessment, conferences and partition details.

Feedback of SEM-I

Report :I] The reviewer of the presentation was satisfied by our objective towards implementing the project.

II] Based on the requirements the reviewer asked about software and connectivity that we are going to use for project.

III] Based on the project reviewer asked about the different algorithms that can be used for our proposed system.

IV] Reviewer suggested the limitations of the project time require to execute project and another approach to execute the project.

V] Concept and idea of the project and related work is accepted by reviewer and encouraged for more improvement.

PAPER PUBLISHED DETAILS: Paper Title: - Automation of Nem-Con

PUBLICATION: - INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH, VOLUME 8, ISSUE 12, DECEMBER-2017
Paper Status: - Submitted ISSN 2229-5518, PUBLISHED

• ASSIGNMENT 2

TITLE: - Project workstation selection, installation along with setup and installation report preparations.

Installation of software:

1. When you plug an Arduino board into a Windows 10 computer, a driver should automatically be installed for it.

2. The driver configures the Arduino as a virtual COM port which can be seen in the Device Manager.

3. Open the Device Manager window by right-clicking the Windows 10 start button in the lower left of the screen and then selecting Device Manager on the menu that pops up.

4. In Device Manager, expand Ports (COM & LPT) and you should see a COM port which your Arduino will be as shown in the image below.

5. If you have other COM ports on the PC, then unplug the Arduino to see which COM port disappears from the device manager, then plug the Arduino back in and see which port appears – this will be the Arduino port, e.g. COM4

Installing Arduino Drivers

4. In the Device Manager, right-click the COM port that was identified as the Arduino (the Arduino must be plugged into the computer first). Select **Update Driver Software...** from the menu that pops up
5. After the driver has been installed, a dialog box will appear that shows that the driver has been successfully updated and will display the Arduino COM port number.
6. Finally back in the Device Manager, the COM port is now identified as an Arduino when the Arduino driver from the Arduino IDE folder is installed.

12.2 Un-installation:

On windows: Arduino IDE 1.6.5r5 and

previous:\Users\(username)\AppData\Roaming\Arduino15
ArduinoIDE1.6.6a and

later:\Users\(username)\AppData\Local\Arduino15

1. If you downloaded Admin Installation as called Windows Installer, you need to delete Arduino IDE from Control Panel.

2. If you downloaded non-Admin Installation, just delete the Arduino setup folder.

12.3 User help

4. The robot is mainly two components an automated component(which is the automated robot AR) and a manual component(which is the manual robot).
5. The manual robot is controlled by the player(human) using a PlayStation controller. to pass the shuttle cock-to the automated robot.
6. The automated robot is powered with AI and makes the decision on changing the zones based on the throwing of the shuttle-cock

• ASSIGNMENT 3

TITLE: - Programming of the project functions, interfaces and GUI as per 1st Term-work submission using corrective actions recommended in Term-1 assessment of Term-work**.**

As the project is more oriented towards hardware components and focuses more on robotics there is very less usage of GUI.

• ASSIGNMENT 4

Title:- To test the project by using automation testing or manual testing.

*** Test Automation for Robot:**

Robot testing is a set of activities conducted through scripts with the motive of finding errors in software and hardware components. It deals with tests for the entire component. It helps to enhance the quality of your robot while reducing costs, maximizing ROI, and saving building time of the robot.

The Robot Framework is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It is a keyword-driven testing framework that uses tabular test data syntax.

For robotic Testing, the testing lifecycles involve various mobility

- manipulation
- human-robot interaction
- sensing
- power/endurance

Log is a built-in keyword that logs the given parameter to the test report generated by the Robot Framework.

* Types of Testing in Robotics:

1.Testing

2.Manupiation Testing

3. Sensing Testing

4.Power Testing

5. Functional Testing

6. Security Testing

Chapter 13

Paper Publication/patent/copyright

• **SEMESTER: I 1. Paper Title:** Prediction Models for Real Estate Prices: A Comparative Study

2. Name of the Conference/Journal where paper submitted: ICCUBEA 2017 @978-1-5386-4008-1/17/$31.00 ©2017 IEEE

3. Paper accepted/rejected: Accepted, Published

4. Review comments by reviewer:

5. Corrective actions if any:

• **SEMESTER: II**

1. Paper Title: Paper Title: PID Controllers for Line Tracing Robots

2.Name of the Conference/Journal where paper submitted:
INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH, VOLUME 8, ISSUE 12, DECEMBER-2017
ISSN 2229-5518

3. Paper accepted/rejected: Accepted And Published

4. Review comments by reviewer:

5. Corrective actions if any:

Chapter 14

Information of Project Group Members

1. Name : Aishwarya Murkute

2.Date of Birth : 22nd March 1996

• Gender : Female

• Permanent Address : Datir Niwas, Behind Jay Towers, Kalbhornagar, Chinchwad Pune-19

• E-Mail murkuteaishwarya@gmail.com

• Mobile/Contact No.: 9850955095

• Placement Details: -

Pursuing higher studies: MS in Computer Science (Admitted to California State University)

Papers Published: 2

1) Title: Prediction Models for Real Estate Prices: A Comparative Study, 978-1-5386-4008-1/17/$31.00 ©2017 IEEE

Name of the Conference/Journal where paper submitted: ICCUBEA 2017, IEEE

2)Title: PID Controllers for Line Tracing Robots

Name of the Conference/Journal where paper submitted: INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH, VOLUME 8, ISSUE 12, DECEMBER-2017 , ISSN 2229-5518

1. Name: Rajarshri Godse

2.Date of Birth:10th Jan 1997

• Gender: Male

• Permanent AddressPrem lok park Chinchwad

• E-Mail rdgodse1997@gmail.com

• Mobile/Contact No.: 8308825247

• Placement Details: -Placed

Paper Published

1)Title: PID Controllers for Line Tracing Robots

Name of the Conference/Journal where paper submitted: INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH, VOLUME 8, ISSUE 12, DECEMBER-2017 , ISSN 2229-5518

Chapter 15

Report Documentation

Table 15.1: Report Documentation

Report Code: CS-BE-Project 2017-2018

Report Title : Student Information Automation of Nem-Con

Name	Address	Email-Id	Contact No	Roll No
Aishwarya Murkute	Santosh Niwas ,Behind Jay Towers, Kalbhornagar Chinchwad Pune	murkuteaishwarya@gmail.com	9850955095	BECOB215
Rajarshi Godse		rdgodse1997@gmail.com	8308825247	BECOA133

Year: 2017-2018 Branch: Computer Engineering

Keywords: Principle Robotics, PID Controller types, PID Controller parameter, PID tuning

Type of Report: Final	Report Checked By:	Report Checked Date:	Guide Name: Prof. Reena Kharat	Total Copies: 4