**GESTURE CONTROL CAR REPORT**



**A PROJECT REPORT**

*Submitted by*

**Partha Prabir Mohapatra UID: 20BEM8002**

**Nandan Kumar UID: 20BEM8010**

**Sunny UID:19BEM1047**

**Frank Leonel FotiHamekong UID:19BEM1058**

***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

**MECHATRONICS ENGINEERING**

**Under the Guidance**

**Of**

**(Name of Guide)**

**University Institute of Engineering**

**CHANDIGARH UNIVERSITY**

**NOVEMBER – 2021**

**CHANDIGARH UNIVERSITY**

**BONAFIDE CERTIFICATE**

Certified that this project report **“GESTURE CONTROL ROBOTIC CAR”**

is the bonafide work of “**”**

**Partha Prabir Mohapatra UID: 20BEM8002**

**Nandan Kumar UID: 20BEM8002**

**Sunny UID: 19BEM1047**

**Frank Leonel fotiHamekong UID: 19BEM1058**

who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Signature of the Head of the Department Signature of the Supervisor

**SIGNATURE SIGNATURE**

<<Name>> <<Name>>

**HEAD OF THE DEPARTMENT SUPERVISOR**

<<Academic Designation>>

<<Department>> <<Department>>

<<Full address of the Dept & College >> <<Full address of the Dept & College >>

**CERTIFICATE FOR EVALUATION**

**College Name: University Institute of Engineering**

**Branch: Mechatronics Engineering.**

**Year / Semester: III / V**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Name of the Students**  **Who have done**  **the project** | **Title of the project** | **Name of the**  **Supervisor with**  **Designation** |
| **1.** | **Partha Prabir Mohapatra** | **GESTURE CONTROL ROBOTIC CAR** | **Inderpreet Singh** |
| **2.** | **Nandan Kumar** |
| **3.** | **Sunny** |
| **4.** | **Frank Leonel fotiHamekong** |

The reports of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Engineering degree in Mechatronics Engineering of Chandigarh University were evaluated and confirmed to be the reports of the work done by the above students and then evaluated.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

|  |  |  |
| --- | --- | --- |
| **Sr. N0.** | **Content**  **Candidate Declaration**  **Acknowledgement** | **Page no** |
| 1. | Introduction........................................ 9 |  |
|  |  |  |
| 2.  3.  4.  5.  6.  7.  8. | Methodology……………………………… 10  Objective  Steps Involved 11-23  Block Diagram of System……………………  Description of Parts  Application and advantages of system 23-24  CODE (interface) 25-40  Future scope…………………………… 40  Conclusion………………………… 42  Reference………………………… 43 |  |

**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI**

**CANDIDATE'S DECLARATION**

I “PARTHA” by here declare that the work embodied in this synopsis entitled

**“GESTURE CONTROL ROBOTIC CAR”**

in partial fulfillment of requirements for the award of degree of B.E (MECHATRONICS ENGINEERING) at **CHANDIGARH UNIVERSITY GHARUAN, MOHALI.** The work which is being presented in this synopsis submitted to **Department of Mechatronics Engineering** is an authentic record of bonafied piece of work.

**ACKNOWLEDGEMENT**

We are using this opportunity to express our gratitude to everyone who supported us throughout the project. We are thankful for their aspiring guidance, invaluable constructive, criticism and friendly advice during the project work. We are Sincerely greatful to them for sharing their truthful and illuminating use on a number of issues related to the project

We are highly indebted to “**Mr. Inderpreet Singh”** for the guidance throughout the project without which project wouldn’t be successful.

Thank also goes to our parents and guardians for their kind co-operation and encouragement which helped us in completion of this project.

Our Thank appreciation also goes to our HOD **“Mr. Harjot Singh Gill”** and our colleagues in developing and people who have willingly helped us out with their abilities.

**GROUP MEMBERS:**

**Partha Prabir Mohapatra(20BEM8002)**

**Nandan Kumar (20BEM8010)**

**Sunny(19BEM1047)**

**Frank Leonel fotiHomekong(19BEM1058)**

INTRODUCTION

In Modern World, automation in robots is increased on a high scale. Many types of technology are upcoming, modified in different categories in automation world. Generally, all this robots controlled by using some hardware equipment. Generation wants everything in fast mode, they want to wait for a long time, so also in this era, we have Technology support. Our Great Engineers invent many things, which help us to grow rapidly in life. so, same in this technological era “I and My team members” come with one more Project known as “**GESTURE CONTROL ROBOTIC CAR**”

So, a robot car controlled by a person’s body gestures is new technology upcoming in today’s world. All the movements of this automated car are controlled by different actions performs by user. car movements will be easily controlled. As human actions are involved in controlling movement, this car can be easily controlled.

Sensors are used to detect the gestures of hand and to transmit the data transmitter and receiver module is used. The same movements are interpreted by Arduino Uno and Accelerometer. For, robot to turned efficiently, motors are connected in cross connection pattern to increase the turning efficiency of the automated car. Two different circuits namely Transmitter & Receiver Circuits namely to establish a proper and secured wireless connection between a person and automated car.

As the name itself suggests that, this is automated car controlled by hand gestures of person. For simple movements of car, if a person’s hand a tilted down then car moves forward direction, if tilted above then cars moves in backward direction, if tilted to right then car turns in left direction and if tilted the left direction, then car turned to right direction. First Accelerometer, gyroscope is used. The sensors have 3 axis x, y, z which detect position of a person’s hand and gives an output in range of serials. This data is received by Arduino Lilypad. Micro-controller of Arduino (ATMEGA328P) then computers and forwards data about direction to which car should move. This data is then sent to an ensure the safety. Then the data is forwarded to transmitter module. At a frequency of 433 MHz this data is sent in air in encoded format.

Now at receiver end, this encoded data is captured by receiver module and then data is sent to decoder IC. This decoded data is about which pin to set high so that motor of the automated car can run accordingly. A motor driver IC Board is used which runs the motor in desired direction given by User’s Hand Gestures. All this process is described in a block diagram part.

The Sensor works on power between 1.8V to 3.6V DC and typically consumes just 350microampere of current which become perfectly compatible for Arduino, It has 3 analog outputs for X, Y, Z axis measurements, 2 pins and self-test pin. The analog outputs are Ratio metric, meaning 0g measurement output as same as half of the 3.3V supply voltage.

Methodology

1. Objective: Our Objective is to make this device simple as well as cheap so it can be produced and used for number of purposes. The objective of this project is to make a car that can be controlled by gesture wirelessly.The user isable tocontrol motions of the car by wearing controller glove and performing predefined gestures. It can be also used in many potential applications such as wireless controller car. Car using the accelerometer with the help of human hand tilting. Signals received and hand tilting. Car movement depends upon the signal made by your hand and from a separation.
2. Steps to be involved: Working is divided into two parts:

Transmitter & Receiver Part

* In Transmitter circuit we have an Arduino Lilypad with the accelerometer and with a encoder like our Bluetooth. First, we have to recognize and detect the movements of hand by using Accelerometer MPU6050 and position of hand is identified by using x and y scale. The data is supplied to Arduino Lilypad. Arduino programmed using Arduino (IDE) Software.

Following algorithm is used to detect and analyses the gestures through the accelerometer and in which direction automated robot should move.

* In Receiver circuit generally, we used an Arduino with a Bluetooth and the driver with the motor and some programming to control the motor and its direction, then the transmitter gives the data in air then receiver circuit captured those data and used our Bluetooth function as a decoder. It decodes the data and send 4 pin outputs of high or low digital signal to L298D motor driver board module.

Then the board runs the automated car in desired direction according to the gesture of the user’s hand.

Block Diagram of System

1. Description of Parts:

* ARDUINO UNO: It is a microcontroller board build on ATmega328 which has 14 digital I/o and 6 analog pins. It has everything that needed to support the microcontroller. It is connected with the computer with a USB Cable to get started with the Arduino Uno board. It is flexible and adaptable in hardware and software. It can sense the environmental changes by receiving input from a variety of sensors. It can affect its surroundings by controlling lights, measure and other actuators.
* ACCELEROMTER(MPU6050): It is a divide that allows for measuring the changes in orientation that take place around in a reference axis, this provides information that allows a device to measure its orientation, maintain it or change it. Here we used 6 axis gyroscopes then therefore a sensor with 6 degrees of freedom. Since it has three axis to measure the rate of turn and around three axis to measures the accelerometer. As you say it combines the 3 cartesian coordinate axis accelerometer and that takes gyroscopic measurements along these three axes pitch, roll, yaw.
* BLUETOOTH(HC-06): Thismodule is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its connection is via serial communication which is an easy way to interface with controller on PC. HC-05 Bluetooth modules allows switching mode between master and slave mode which means it free to use neither receiving nor transmitting data.
* BATTERY: Battery is a device which is combination of one or more electrochemical cells. It is a device that directly converts chemical energy to the electrical energy. The purpose of battery is to supply 9-12 volts to operate DC motors.
* DC MOTOR: It is used for the conversion of direct current into mechanical motion. The mechanical principle motion could be rotary or linear. The functioning of this motor is based on that when a current carrying conductor is settled in a magnetic field, the conductor experiences a mechanical force. The speed of a Dc motor can be controlled by changing the voltage applied to armature or by changing the field current. DC motors can be used can be used for the movement of the robotic car.
* MOTOR DRIVER: This L298n is one type ofmotor driver module, Ithasa high-power motor driver module for driving dc and stepper motors. L298N modules can run up to 4 dc motors, or 2 dc motors with directional and speed control. The L298n motor driver module be fabricated with a motor driver IC 78M05 Voltage regulator, resistors, capacitor, power led, 5V jumper in an integrated circuit. M05 voltage regulator will be enabled only when the jumper is placed. At the core of the module is the big, black chip with chunky heat sink is an L298n a dual channel H-Bridge motor driver have the ability of driving a pair of DC motors.
* JUMPER WIRES: Jumper wires are used to connect two points in a circuit. customarily used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Male jumpers are made to plug securely into the holes in a breadboard. Female jumpers are used for combining male header posts and pin terminals on components. Jumpers are applicable in female-female, male-male and male-female configurations.
* SOFTWARE DEVELOPMENT: ARDUINO (IDE) software has been used as the interface between software and hardware of this project. It needs a program to operate and execute the process associated with proposed design. The process is easy to verify and compile after writing the code. Here, it connects to the ARDUINO hardware to upload programs. But before upload the program there is a need to select appropriate microcontroller like ARDUINO UNO and ARDUNO LILYPAD from the Tool menu has been chosen. And for proper communication with computer and ARDUINO boards there is a need to select COM port from the tool menu.

1. **DESIGN AND CONSTRUCTION**

In this section of our paper, the different components used in our project are explained. As already mentioned, the project mainly is divided into the transmitter and receiver circuit. The parts of each circuit are given below.

Diagram

Description automatically generated

**FOR TRANSMITTER**

* Arduino Lilypad 328
* HC-05 Bluetooth module
* MPU6050 Gyro with accelerometer
* Battery (9-12V)
* Hand gloves

**FOR RECEIVER**

* Arduino Uno
* HC-05 Bluetooth module
* L298N H-Bridge motor driver
* Robot car chassis, DC motor and wheels
* Battery (9-12V)

**Arduino UNO**

Arduino is an open-source electronics platform based on easy-to-use hardware and software able to read inputs, light on a sensor, control a motor and many more applications. It is the brain of the car and is installed with some code. Instructions are sent to the devices connected to the board by use of the Arduino programming language or the Arduino IDE.

The board itself is made of different pins that serve various functions and can be used as of the wish of the programmer in order to complete any project.It has everything that is needed to support the microcontroller. Simply connect it to the computer with a USB cable to get started with the Arduino board. It is flexible, easy to use hardware and software.

The Arduino UNO is a microcontroller board based on the Atmega328P. It has 14 digital I/O pins (Of which 6 are used as PMW output pins). It also has 6 analog input pins.

**HC-05**

HC-05 Bluetooth Module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data. It uses the 2.45GHz frequency band. The transfer rate of the data can vary up to 1Mbps and is in range of 10 meters.

Graphical user interface, diagram

Description automatically generated

**MPU6050**

MPU6050 is a Micro Electro-Mechanical System (MEMS) that consists of three-axis accelerometer and three-axis gyroscope. It helps us to measure velocity, orientation, acceleration, displacement and other motion like features.

It consists of a 16-bit analog to digital converter hardware which enables it to capture three-dimension motion at the same time.Here we used 6 axis gyroscopes then therefore a sensor with 6 degrees of freedom, since it has three axes to measure the rate of turn and another three axis to measures the acceleration.

A picture containing text, electronics, circuit

Description automatically generated

**L298N**

L298N module is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. It can control both the speed and rotation direction of two DC motors. This module consists of an L298 dual-channel H-bridge motor driver IC. This module uses two techniques for the control of the speed and rotation direction of the motors. These are PMW for controlling the speed and H-Bridge for controlling rotation direction. These modules can control two DC motors or one stepper motor at the same time.

Graphical user interface, schematic

Description automatically generated

**DC Motor**

DC motor is used for the conversion of direct current into mechanical motion. The mechanical motion could be rotary or linear. The operation of DC motor is based on the principle that when a current carrying conductor is placed in a magnetic field, the conductor experiences a mechanical force. The speed of a DC motor can be controlled by changing the voltage applied to the armature or by changing the field current. DC motors can be used for the movement of the robotic car.



**Battery**

A battery is a device consisting of one or more electrochemical cells. A battery is device that directly converts chemical energy to the electrical energy. The purpose of battery is to supply 9 -12 volts to operate DC motors.



1. **SOFTWARE**
2. **Coding**

The microcontroller is programmed such that when a certain range of tilting occurs, it will transmit signals to the motor to move to the direction required.

Arduino (IDE)software has been used as the interface between software and hardware of this project. The microcontroller needs a program to operate and execute the process associated with proposed design. It is easy to check and compile after writing the code.

Here, it connects to the Arduino hardware to upload programs. But before upload the program there is a need to select appropriate microcontroller like Arduino UNO and Arduino Lilypad from the Tool menu has been chosen. And for proper communication with computer and Arduino boards there is a need to select COM port from the tool menu.

1. **Experimental Setup**

A picture containing text, electronics, screenshot

Description automatically generated

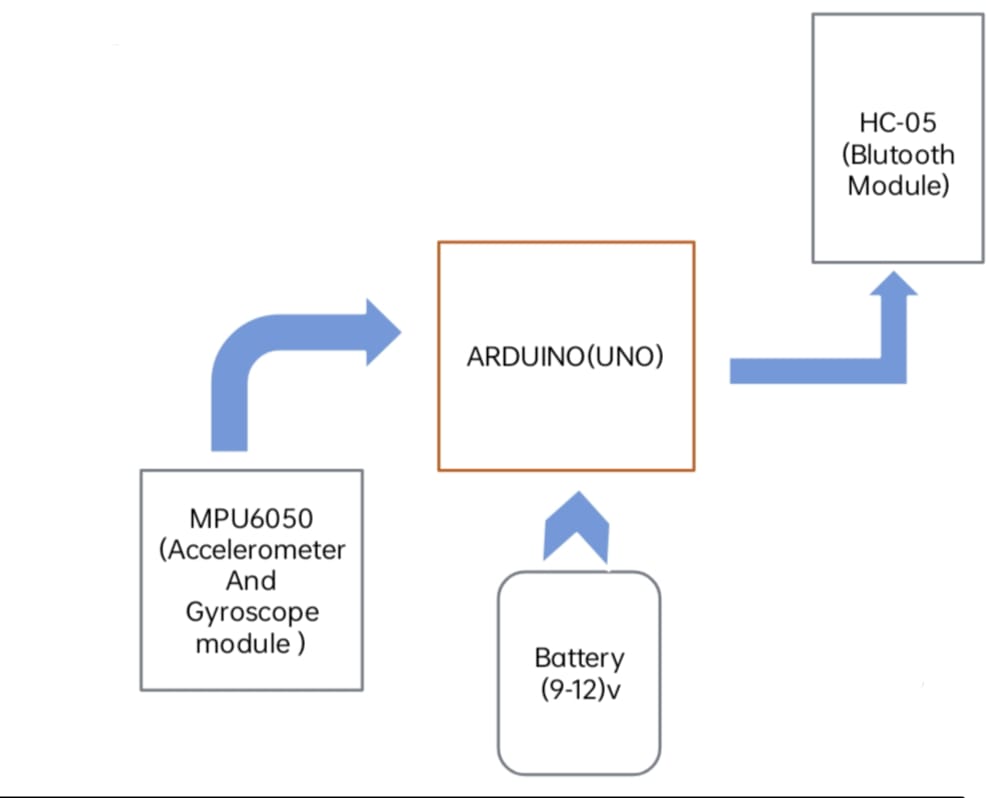
**Working of Transmitter**

In transmitter circuit we have an Arduino Lilypad with the accelerometer and with an encoder like our Bluetooth module.

Here first step is to recognize and detect the movements of hand by using Accelerometer MPU6050 and the position of hand is identified by using x and y scale. This data is supplied to Arduino Lilypad. Arduino has a micro controller ATMEGA 328 which can be programmed using Arduino (IDE) Software.

Following algorithm is used to detect and analyses the gestures through the accelerometer and in which direction automated robot should move.

+

****

**Working of Receiver**

In receiver circuit, generally we use an Arduino with a Bluetooth module and the driver with the motor and some programming to control the motor and its direction. When the transmitter circuit gives the data in air then the receiver circuit captured those data and used our Bluetooth function as a decoder.

It decodes the data and sends 4 pin outputs of high or low digital signal to L298D motor driver board module.

And then the board runs the automated car in desired direction according to the gesture of the user’s hand

Diagram

Description automatically generated

Application and advantages of system:

Advantages:

* Simple, fast and easy to implement and can be applied on real system.
* Speed and sufficient reliable for recognition system
* It is very much compact and user friendly
* It has low power consumption

Applications:

* The use of gesture recognition, remote Control with the wave of a hand of various devices is possible such as gaming to make the game player’s experience more interactive and impressive
* Gesture controlling is very helpful for handicapped and physically disabled people to do certain tasks, such as driving a vehicle.
* Wireless controlled robots are very useful in many applications like remote surveillance, military, medical etc.
* Hand gesture-controlled robot can be utilized used by physically challenged in wheelchair.

**CODE:**

**RECIEVER PART:-**

**#include <SoftwareSerial.h>**

**SoftwareSerialBTSerial(2, 3); //**

**char tiltDirection;**

**int motorInput1 = 11;**

**int motorInput2 = 10;**

**int motorInput3 = 9;**

**int motorInput4 = 8;**

**void setup() {**

**pinMode(motorInput1, OUTPUT);**

**pinMode(motorInput2, OUTPUT);**

**pinMode(motorInput3, OUTPUT);**

**pinMode(motorInput4, OUTPUT);**

**digitalWrite(motorInput1, LOW);**

**digitalWrite(motorInput2, LOW);**

**digitalWrite(motorInput3, LOW);**

**digitalWrite(motorInput4, LOW);**

**Serial.begin(115200); // Serial communication is activated at 38400 baud/s.**

**BTSerial.begin(38400); // HC-05 default speed in AT command more**

**}**

**void loop() {**

**if (BTSerial.available()) { // Witing for data incoming from the other XBee module**

**tiltDirection = BTSerial.read();**

**if(tiltDirection == 'F'){**

**Serial.println("Backward");**

**}else if(tiltDirection == 'B'){**

**Serial.println("Forward");**

**forward();**

**}else if(tiltDirection == 'R'){**

**Serial.println("Left");**

**left();**

**}else if(tiltDirection == 'L'){**

**Serial.println("Right");**

**right();**

**}else if(tiltDirection == 'S'){**

**Serial.println("Stop");**

**stopCar();**

**}**

**}**

**}**

**void forward()**

**{**

**/\*The pin numbers and high, low values might be different depending on your connections \*/**

**digitalWrite(motorInput1, LOW);**

**digitalWrite(motorInput2, HIGH);**

**digitalWrite(motorInput3, LOW);**

**digitalWrite(motorInput4, HIGH);**

**}**

**void reverse()**

**{**

**/\*The pin numbers and high, low values might be different depending on your connections \*/**

**digitalWrite(motorInput1, HIGH);**

**digitalWrite(motorInput2, LOW);**

**digitalWrite(motorInput3, HIGH);**

**digitalWrite(motorInput4, LOW);**

**}**

**void right()**

**{**

**/\*The pin numbers and high, low values might be different depending on your connections \*/**

**digitalWrite(motorInput1, LOW);**

**digitalWrite(motorInput2, HIGH);**

**digitalWrite(motorInput3, LOW);**

**digitalWrite(motorInput4, LOW);**

**}**

**void left()**

**{**

**/\*The pin numbers and high, low values might be different depending on your connections \*/**

**digitalWrite(motorInput1, LOW);**

**digitalWrite(motorInput2, LOW);**

**digitalWrite(motorInput3, LOW);**

**digitalWrite(motorInput4, HIGH);**

**}**

**void stopCar() {**

**digitalWrite(motorInput1, LOW);**

**digitalWrite(motorInput2, LOW);**

**digitalWrite(motorInput3, LOW);**

**digitalWrite(motorInput4, LOW);**

**}**

**TRANSMITTER PART:-**

**#include "I2Cdev.h"**

**#include "MPU6050\_6Axis\_MotionApps20.h"**

**// Arduino Wire library is required if I2Cdev I2CDEV\_ARDUINO\_WIRE implementationis used in I2Cdev.h**

**#include "Wire.h"**

**#include <SoftwareSerial.h>**

**SoftwareSerialBTSerial(10, 11); // CONNECT BT RX PIN TO ARDUINO 11 PIN | CONNECT BT TX PIN TO ARDUINO 10 PIN**

**#define OUTPUT\_READABLE\_YAWPITCHROLL**

**#define INTERRUPT\_PIN 2 // use pin 2 on Arduino Uno & most boards**

**#define LED\_PIN 13 // (Arduino is 13, Teensy is 11, Teensy++ is 6)**

**MPU6050 mpu;**

**bool blinkState = false;**

**// MPU control/status vars**

**bool dmpReady = false; // set true if DMP init was successful**

**uint8\_t mpuIntStatus; // holds actual interrupt status byte from MPU**

**uint8\_t devStatus; // return status after each device operation (0 = success, !0 = error)**

**uint16\_t packetSize; // expected DMP packet size (default is 42 bytes)**

**uint16\_t fifoCount; // count of all bytes currently in FIFO**

**uint8\_t fifoBuffer[64]; // FIFO storage buffer**

**// orientation/motion vars**

**Quaternion q; // [w, x, y, z] quaternion container**

**VectorFloatgravity; // [x, y, z] gravity vector**

**float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector**

**float pitch = 0;**

**float roll = 0;**

**float yaw = 0;**

**int x;**

**int y;**

**// ================================================================**

**// === INTERRUPT DETECTION ROUTINE ===**

**// ================================================================**

**volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has gone high**

**void dmpDataReady() {**

**mpuInterrupt = true;**

**}**

**// ================================================================**

**// === INITIAL SETUP ===**

**// ================================================================**

**void setup() {**

**// join I2C bus (I2Cdev library doesn't do this automatically)**

**#if I2CDEV\_IMPLEMENTATION == I2CDEV\_ARDUINO\_WIRE**

**Wire.begin();**

**Wire.setClock(400000); // 400kHz I2C clock. Comment this line if having compilation difficulties**

**#elif I2CDEV\_IMPLEMENTATION == I2CDEV\_BUILTIN\_FASTWIRE**

**Fastwire::setup(400, true);**

**#endif**

**// initialize serial communication**

**// (115200 chosen because it is required for Teapot Demo output, but it's**

**// really up to you depending on your project)**

**Serial.begin(115200);**

**BTSerial.begin(38400); // HC-05 default speed in AT command more**

**while (!Serial); // wait for Leonardo enumeration, others continue immediately**

**// NOTE: 8MHz or slower host processors, like the Teensy @ 3.3V or Arduino**

**// Pro Mini running at 3.3V, cannot handle this baud rate reliably due to**

**// the baud timing being too misaligned with processor ticks. You must use**

**// 38400 or slower in these cases, or use some kind of external separate**

**// crystal solution for the UART timer.**

**// initialize device**

**Serial.println(F("Initializing I2C devices..."));**

**mpu.initialize();**

**pinMode(INTERRUPT\_PIN, INPUT);**

**// verify connection**

**Serial.println(F("Testing device connections..."));**

**Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));**

**// load and configure the DMP**

**Serial.println(F("Initializing DMP..."));**

**devStatus = mpu.dmpInitialize();**

**// supply your own gyro offsets here, scaled for min sensitivity**

**mpu.setXGyroOffset(126);**

**mpu.setYGyroOffset(57);**

**mpu.setZGyroOffset(-69);**

**mpu.setZAccelOffset(1869); // 1688 factory default for my test chip**

**// make sure it worked (returns 0 if so)**

**if (devStatus == 0) {**

**// turn on the DMP, now that it's ready**

**Serial.println(F("Enabling DMP..."));**

**mpu.setDMPEnabled(true);**

**// enable Arduino interrupt detection**

**Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));**

**attachInterrupt(digitalPinToInterrupt(INTERRUPT\_PIN), dmpDataReady, RISING);**

**mpuIntStatus = mpu.getIntStatus();**

**// set our DMP Ready flag so the main loop() function knows it's okay to use it**

**Serial.println(F("DMP ready! Waiting for first interrupt..."));**

**dmpReady = true;**

**// get expected DMP packet size for later comparison**

**packetSize = mpu.dmpGetFIFOPacketSize();**

**} else {**

**// ERROR!**

**// 1 = initial memory load failed**

**// 2 = DMP configuration updates failed**

**// (if it's going to break, usually the code will be 1)**

**Serial.print(F("DMP Initialization failed (code "));**

**Serial.print(devStatus);**

**Serial.println(F(")"));**

**}**

**// configure LED for output**

**pinMode(LED\_PIN, OUTPUT);**

**}**

**// ================================================================**

**// === MAIN PROGRAM LOOP ===**

**// ================================================================**

**void loop() {**

**// if programming failed, don't try to do anything**

**if (!dmpReady) return;**

**// wait for MPU interrupt or extra packet(s) available**

**while (!mpuInterrupt&&fifoCount<packetSize) {**

**// other program behavior stuff here**

**// .**

**// .**

**// .**

**// if you are really paranoid you can frequently test in between other**

**// stuff to see if mpuInterrupt is true, and if so, "break;" from the**

**// while() loop to immediately process the MPU data**

**// .**

**// .**

**// .**

**}**

**// reset interrupt flag and get INT\_STATUS byte**

**mpuInterrupt = false;**

**mpuIntStatus = mpu.getIntStatus();**

**// get current FIFO count**

**fifoCount = mpu.getFIFOCount();**

**// check for overflow (this should never happen unless our code is too inefficient)**

**if ((mpuIntStatus& 0x10) || fifoCount == 1024) {**

**// reset so we can continue cleanly**

**mpu.resetFIFO();**

**Serial.println(F("FIFO overflow!"));**

**// otherwise, check for DMP data ready interrupt (this should happen frequently)**

**} else if (mpuIntStatus& 0x02) {**

**// wait for correct available data length, should be a VERY short wait**

**while (fifoCount<packetSize) fifoCount = mpu.getFIFOCount();**

**// read a packet from FIFO**

**mpu.getFIFOBytes(fifoBuffer, packetSize);**

**// track FIFO count here in case there is > 1 packet available**

**// (this lets us immediately read more without waiting for an interrupt)**

**fifoCount -= packetSize;**

**#ifdef OUTPUT\_READABLE\_YAWPITCHROLL**

**// display Euler angles in degrees**

**mpu.dmpGetQuaternion(&q, fifoBuffer);**

**mpu.dmpGetGravity(&gravity, &q);**

**mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);**

**yaw = ypr[0] \* 180 / M\_PI;**

**pitch = ypr[1] \* 180 / M\_PI;**

**roll = ypr[2] \* 180 / M\_PI;**

**if (roll > -100 && roll < 100)**

**x = map (roll, -100, 100, 0, 100);**

**if (pitch > -100 && pitch < 100)**

**y = map (pitch, -100, 100, 100, 200);**

**Serial.print(x);**

**Serial.print("\t");**

**Serial.println(y);**

**if((x>=45 && x<=55) && (y>=145 && y <=155)){**

**BTSerial.write('S');**

**}else if(x>60){**

**BTSerial.write('R');**

**}else if(x<40){**

**BTSerial.write('L');**

**}else if(y>160){**

**BTSerial.write('B');**

**}else if(y<140){**

**BTSerial.write('F');**

**}**

**#endif**

**// blink LED to indicate activity**

**blinkState= !blinkState;**

**digitalWrite(LED\_PIN, blinkState);**

**}**

**}**

**FUTURE SCOPE**

The concept of hand gesture robot may have applications in hazardous areas out of the reach of humans in order to ensure the safety of workers whereby the robot controlled at a distance, under human supervision will perform the task in hand, A typical example will be carrying heavy objects in factories. Here, humans are replaced with such hand gestured cars to perform the tasks. This technology enables large movements by just making small hand gestures.

The devices which are controlled by gestures will work efficiently in the field of Defense, industry, medial, etc. The future doctors will look into the human body with image manipulation or can move the tiny robot in the body using gestures and can find the problem,. This will help physically challenged people as they can move certain objects with less physical movement.

**CONCLUSION**

In this paper, a design strategy for controlling a robot car based on Arduino Uno microcontroller

has been demonstrated, which can be programmed to react to events based on hand gesture and to cause corresponding actions. The mechanism involves controlling the robot car based on hand gesture recognition (car moves similarly to the direction and position of hand) with the transmission of data done by HC-05 Bluetooth module. This project can be helpful where the humans are not able to perform any task but can be defined or the task with some device using hand gestures.

**REFERENCES**

[1] Yanan Chang, “Dynamic gesture recognition based on HMM,” MPhil.

thesis, South China University of Technology, Guangzhou, GD, China,

2012.

[2] Hang Zhao, “Motion control design of gesture recognition for mobile

robot,” M.S. thesis, University of Electronic Science and Technology,

Chengdu, SC, China, 2017.

[3] Chen Gao, “Research on the control of mobile robots based on dynamic

and static gestures,” M.S. thesis, Beijing University of Chemical

Technology, Nanjing, JS, China, 2017.

[4] Gould S J J, Brumby D P, Cox A L, et al., “Methods for Human Computer Interaction Research,” in The ACM Conference Extended.

ACM, 2015, pp. 2473-2474.

[5] Tianxing Wang, “Gesture-based remote-control technology for

unmanned platforms,” M.S. thesis, Nanjing University, JS, China, 2017.

[6] V. Bonato, A. K. Sanches, Fernandes M M, et al., “A Real Time Gesture

Recognition System for Mobile Robots,” in Icinco 2004, Proceedings of

the First International Conference on Informatics in Control, Automation

and Robotics, Setubal, Portugal, August. DBLP, 2004, pp. 207-214.

[7] RM. Sanso, D. Thalmann, “A Hand Control and Automatic Grasping

System for Synthetic Actors,” Computer Graphics Forum, vol. 13, no. 3,

pp. 167-177, Aug. 1994.

[8] Robert Y. Wang, “Real-time hand-traking with a color glove,” in ACM

SIGGRAPH.ACM, 2009, pp. 1-8.

[9] Poularakis S, Katsavounidis I, “Finger detection and hand posture recognition based on depth information,” in IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE,2014, pp. 4329-4333. [10] Yunde Zhang, “Research and application of gesture recognition technology,” M.S. thesis, Anhui University, Hefei, AH, China, 2013. [11] Yuanxin Zhu, Guangyou Xu, “Appearance-based dynamic isolated gesture recognition,” Journal of Software, vol. 11, no.1, pp. 54-61, Jan.2000 [12] Xueping Zhao, Jianming Wang, Haiting Liu, et al., “Research on Recognition System of Indoor Robot Dynamic Gesture Command,” Computer Engineering and Applications, vol. 47, no. 33, pp. 209-212, Nov. 2011.