**Wireless Gesture Controlled Robot Using Accelerometer & Arduino**

**Introduction**

The Wireless Gesture Controlled Robot is an embedded system designed to assist individuals with disabilities in moving objects using head gestures. The system utilizes an accelerometer sensor, Arduino Nano as a remote control (master), Arduino Uno as the actual robot car (slave), and HC-05 Bluetooth modules to establish a wireless communication link between the two devices. The accelerometer sensor detects head movements, which are wirelessly transmitted to the robot car, enabling it to perform specific actions.

This documentation provides a step-by-step guide on how to set up and operate the Wireless Gesture Controlled Robot, including the hardware components, circuit connections, and necessary software programming.

**Hardware Components**

*To build the Wireless Gesture Controlled Robot, you will need the following hardware components.*

1. Arduino Nano (Master/Remote Control)

2. Arduino Uno (Slave/Actual Car)

3. Accelerometer Sensor (e.g., MPU6050)

4. HC-05 Bluetooth Modules (2)

5. Motor Driver Module (e.g., L298N)

6. Robot Car Chassis with Motors and Wheels

7. Power Supply (Batteries or DC Adapter)

8. Jumper Wires

9. Breadboard (optional for prototyping)

**Circuit Connections**

Follow the instructions below to establish the circuit connections for the Wireless Gesture Controlled Robot

***Arduino Nano (Master/Remote Control) Connections***

1. Connect the VCC pin of the accelerometer sensor to the 3.3V pin of the Arduino Nano.

2. Connect the GND pin of the accelerometer sensor to the GND pin of the Arduino Nano.

3. Connect the SDA pin of the accelerometer sensor to the A4 pin of the Arduino Nano.

4. Connect the SCL pin of the accelerometer sensor to the A5 pin of the Arduino Nano.

5. Connect the TX pin of the Arduino Nano to the RX pin of the HC-05 Bluetooth module.

6. Connect the RX pin of the Arduino Nano to the TX pin of the HC-05 Bluetooth module.

7. Connect the VCC and GND pins of the HC-05 Bluetooth module to the appropriate power supply pins.

***Arduino Uno (Slave/Actual Car) Connections***

1. Connect the motor driver module to the Arduino Uno following the module's specifications.

2. Connect the left motor terminals of the robot car to the motor output pins of the motor driver module.

3. Connect the right motor terminals of the robot car to the motor output pins of the motor driver module.

4. Connect the VCC and GND pins of the motor driver module to the appropriate power supply pins.

5. Connect the TX pin of the Arduino Uno to the RX pin of the HC-05 Bluetooth module.

6. Connect the RX pin of the Arduino Uno to the TX pin of the HC-05 Bluetooth module.

7. Connect the VCC and GND pins of the HC-05 Bluetooth module to the appropriate power supply pins.

*Note: Ensure that the power supply used for the motors is appropriate for their voltage and current requirements.*

**Software Programming**

To program the Wireless Gesture Controlled Robot, follow these steps:

1. Install the Arduino IDE (Integrated Development Environment) on your computer if you haven't already.

2. Connect the Arduino Nano (Master/Remote Control) and the Arduino Uno (Slave/Actual Car) to your computer

*using separate USB cables.*

3. Open the Arduino IDE and create a new sketch for the Arduino Nano.

4. Install the necessary libraries for the MPU6050 accelerometer sensor. Go to Sketch - Include Library - Manage Libraries, then search for MPU6050 and install the library.

5. Copy and paste the following code into the Arduino IDE for the Arduino Nano

#include <Wire.h>

#include "accelerometer.h"

//==============================================================

int White = 5;

int Data = 0;

//==============================================================

void printData()

{ Serial.write ("X=");

Serial.println(accelX);

Serial.write ("Y=");

Serial.println(accelY);

Serial.write ("Z=");

Serial.println(accelZ);

Serial.println();

}

//==============================================================

void setup() {

Serial.begin (9600);

Wire.begin();

setupMPU();

}

//==============================================================

void loop() {

recordAccelRegisters();

printData();

//==============================================================

if (accelX > 11000)

{

Serial.println ('B');

}

else if (accelX < -5000)

{

Serial.println ('F');

}

else if (accelY > 11000)

{

Serial.println ('R');

}

else if (accelY < -11000)

{

Serial.println ('L');

}

else

{

Serial.println ('S');

}

  delay(500);

}

6. Customize the code according to your specific robot car's motor control mechanism. Modify the conditions inside the if statements or switch-case structure to control the motors appropriately based on the accelerometer data.

7. Verify and upload the code to the Arduino Nano (Master/Remote Control).

8. Disconnect the Arduino Nano from your computer and connect it to the power supply (batteries or DC adapter) for standalone operation.

9. Open another instance of the Arduino IDE and create a new sketch for the Arduino Uno (Slave/Actual Car).

10. Copy and paste the following code into the Arduino IDE for the Arduino Uno

int Data = 0;

int Motor1A = 5;

int Motor1B = 6;

int Motor2A = 9;

int Motor2B = 10;

//==============================================================

void forward (){

digitalWrite (Motor1A, HIGH);

digitalWrite (Motor2A, HIGH);

digitalWrite (Motor1B, LOW);

digitalWrite (Motor2B, LOW);

}

void backward (){

digitalWrite (Motor1A, LOW);

digitalWrite (Motor2A, LOW);

digitalWrite (Motor1B, HIGH);

digitalWrite(Motor2B, HIGH);

}

void left (){

digitalWrite (Motor1A, HIGH);

digitalWrite (Motor2A, LOW);

digitalWrite (Motor1B, LOW);

digitalWrite(Motor2B, LOW);

}

void right (){

digitalWrite (Motor1A, LOW);

digitalWrite (Motor2A, HIGH);

digitalWrite (Motor1B, LOW);

digitalWrite(Motor2B, LOW);

}

void stay (){

digitalWrite (Motor1A, LOW);

digitalWrite (Motor2A, LOW);

digitalWrite (Motor1B, LOW);

digitalWrite(Motor2B, LOW);

}

//==============================================================

void setup() {

Serial.begin (9600);

pinMode (Motor1A, OUTPUT);

pinMode (Motor2A, OUTPUT);

pinMode (Motor1B, OUTPUT);

pinMode (Motor2B, OUTPUT);

}

//==============================================================

void loop() {

if (Serial.available() > 0)

{

Data = Serial.read();

}

if (Data == 'F')

{

forward();

}

else if (Data == 'B')

{

backward();

}

else if (Data == 'L')

{

left();

}

else if (Data == 'R')

{

right();

}

else if (Data == 'S')

{

   stay();

  }

}

11. Customize the code in the Arduino Uno sketch to control the motors according to the signals received from the Arduino Nano. Modify the motor control logic based on the specific motor driver module and connections used in your robot car.

12. Verify and upload the code to the Arduino Uno (Slave/Actual Car).

13. Disconnect the Arduino Uno from your computer and connect both the Arduino Nano (Master/Remote Control) and the Arduino Uno (Slave/Actual Car) to their respective power supplies (batteries or DC adapters).

**Usage**

Once you have completed the hardware setup and uploaded the code to the Arduino boards, follow these steps to use the Wireless Gesture Controlled Robot

1. Turn on the power supply for both the Arduino Nano (Master/Remote Control) and the Arduino Uno (Slave/Actual Car).

2. Hold the Arduino Nano in a suitable position, such that the accelerometer sensor can detect your head gestures.

3. Tilt your head in the desired direction to control the robot car's movements.

4. The accelerometer data from the Arduino Nano will be wirelessly transmitted via the HC-05 Bluetooth module to the Arduino Uno.

5. The Arduino Uno (Slave/Actual Car) will receive the signals and control the motors accordingly.

*Note: Ensure that the HC-05 Bluetooth modules are properly paired and connected to each Arduino board. Adjust the Bluetooth module settings (such as baud rate) if necessary.*

**Conclusion**

The Wireless Gesture Controlled Robot offers a user-friendly and innovative solution for individuals with disabilities, allowing them to manipulate objects using head gestures. By following this documentation, you should be able to build, program, and operate the Wireless Gesture Controlled Robot using Arduino Nano (Master/Remote Control) and Arduino Uno (Slave/Actual Car) with HC-05 Bluetooth modules. Remember to take safety precautions while operating the robot car, and feel free to experiment and enhance the system based on your specific requirements.