**Project Report**

**Automatic Wiper System**

This project addresses the automatic wiper and car security system for a car. Apart from that, it takes care of the security in terms of Smartphone App based monitoring and control of the car door.

The innovation is in terms of use of Arduino Nano for Automatic activation of the wiper when rain falls as well as the speed control of the same under heavy or light rain conditions. Also the [PCB](https://electronicsforu.com/videos-slideshows/videos/make-pcb-quickly) is a double sided compact sized one implemented with Eagle software.

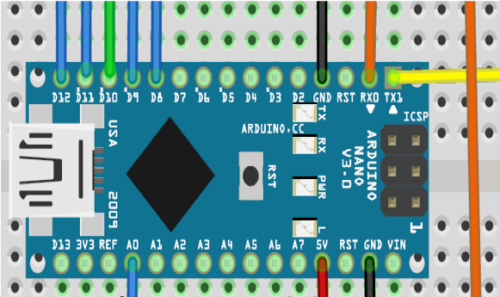
Here in this project, the wiper system is a single motor based wiper which wipes the Car Windshield at once. It also changes its speed automatically depending on the intensity of rain falling.

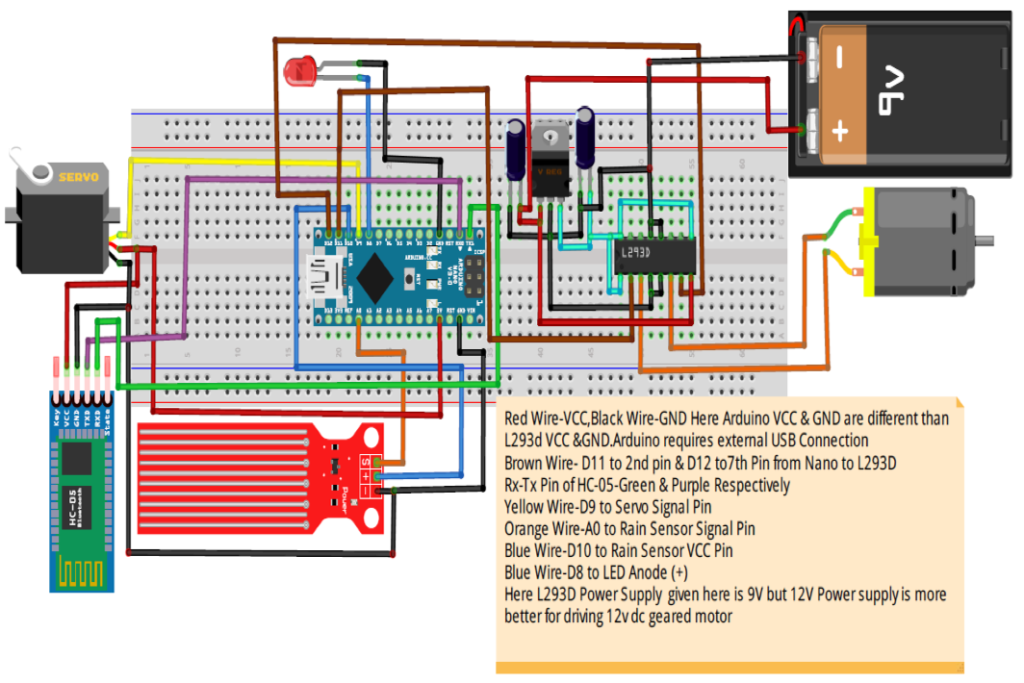
List of Components used for automatic wiper and car security system:

|  |  |  |  |
| --- | --- | --- | --- |
| **+.** | **Components** | **Quantity** | **Technical Specifications** |
| 1 | Arduino Nano v3.0 | 1 | Any Arduino Family Microcontroller is applicable |
| 2 | Servo Motor | 1 | SG-90 Servo Motor |
| 3 | Water/Rain Sensor | 1 | Senses the water and act as a Conducting Switch |
| 4 | HC-05 Bluetooth Module | 1 | Range upto 10m with Frequency of 2.4 GHz and easily connected to Smartphone devices |
| 5 | L293D | 1 | H-Bridge Motor Driver to Drive Motor |
| 6 | IC 7805 | 1 | Regulate the Input Voltage to 5V as its Output |
| 7 | Motor | 1 | 12V DC Geared Motor |
| 8 | 10uf Capacitor | 2 | 10ufarad 63V capacity |
| 9 | LED | 3 |  |
| 10 | Screw Clamp | 1 | Wago Screw Clamp 2 port Connector |
| 11 | 12V DC Jack | 1 | DC Jack for connecting the 12V Adapter Pin- |
| 12 | Jumper Wires |  | – |
| 13 | Batteries/ Adapter | 1 | 12V Adapter or 12V Battery Pack is used |

Circuit Diagram & Connections

The image shows the whole Circuit Diagram of the Project. The Important Element of this project is the Arduino Nano v3.0 Microcontroller. As this project is compatible with any Arduino Family Microcontroller, but due to small size and compact body we preferred Arduino Nano. The Arduino Nano is connected with the other components which also plays an important role in this project. These components are the Rain Sensor, HC-05 Bluetooth Module, SG-90 Servo Motor & L293D H-Bridge Motor Driver IC.  
Arduino Nano with the interfaced components. Arduino Nano dimension is 18mm x 45mm. Final PCB dimension is 50mm x 50mm.

****



**Software Code Algorithm**

#define Rain\_sensor A0 // Analog pin A0 of Arduino Nano  
int Motorinp-1=11;  
int Motorinp-2=12;  
int Engine=8;  
int Sensorinput=10;  
int Servo Input=9;  
int inp=0; // Value which connects the Unlock Door Loop with Engine\_On\_Off loop//  
int rs=0; // Value which connects the Rain Sensor loop with Engine\_Off loop//  
String data; //Read Serial Data//  
int pos=0; // define Servo Position//  
int val; // Read Analog Value//  
————————————————————————————————————————–  
#include  
#include  
————————————————————————————————————————–  
void setup()  
{  
(Motorinp-1, OUTPUT);  
(Motorinp-2, OUTPUT);  
(Sensorinput, OUTPUT);  
(Engine, OUTPUT); (  
Rain\_sensor, INPUT); // Defining the pinModes//  
Serial.begin(9600); // Baud Rate//  
mySerial.begin(9600); // Serial Read takes place at 9600 baud rate per second//  
}  
————————————————————————————————————————–  
void loop()  
{ while (mySerial.available())  
{ delay(10 );  
char c = mySerial.read();  
if (c == ‘#’)  
{ break; }  
data += c; }  
————————————————————————————————————————–  
if (data.length() > 0)  
{  
If (data == “a”) // If the Serial Read value is a than Unlock \_Car \_Door loop gets active. //  
{ Unlock\_Car\_Door(); // Car Door gets unlock//  
inp=1; // This value will activate the Engine\_On & Engine\_Off loop// }  
else if(data == “b”) //If the Serial Read value is b than Lock\_Car \_Door loop active. //  
{ Lock\_Car\_Door(); // Car Door gets lock//  
inp=0;  
Engine\_Off(); // Car system gets deactivate// }  
else if(data == “c” && inp == 1 ) //Unlock\_Car\_Door loop activates Engine\_On Loop. //  
{ Engine\_On(); // Car system gets deactivate// }  
else if(data == “d” && inp == 1 ) //Unlock\_Car\_Door loop activates Engine\_Off Loop. //  
{ Engine\_Off(); // Car system gets deactivate// } }  
if(inp==1 && rs==1) // Unlock\_Car\_Door & Engine\_ON loop activates Engine\_Off ///  
{ RainSensor(); }  
data=””; }  
————————————————————————————————————————–  
void Unlock\_Car\_Door()  
{ myservo.write(pos); // Door will move in Clockwise Direction//  
delay(z); // On basis of requirement of how fast door should work// }  
void Lock\_Car\_Door()  
{ myservo.write(pos); //Door will move in Anticlockwise Direction//  
delay(z); // On basis of requirement of how fast door should work// }  
void Engine\_On()  
{ Engine pin gets HIGH // digital output//  
Sensorinput pin gets HIGH // digital inputs//  
RainSensor(); // RainSensor Loop//  
rs=1; // Value which connects the Rain Sensor loop with Engine\_On loop// }  
void Engine\_Off() // Engine Off loop//  
{ Engine pin gets LOW // digital output//  
rs=0; //// Value which connects the Rain Sensor loop with Engine\_Off loop//  
Sensorinput getsLOW); //digital input//  
Motorinp-1 pin gets LOW // digital output//  
Motorinp-2 pin gets LOW // digital output// }  
————————————————————————————————————————–  
void RainSensor() // Rain Sensor Loop//  
{  
val= analogRead(Rain\_sensor); // Rain Sensor Values//  
if(val>x1 && val<x2) //Rain Sensor Values such that x2>x1//  
{ Motorinp -1 pin gets HIGH // digital output//  
Motorinp-2 pin gets LOW // digital output//  
delay(y1);  
Motorinp-2 pin gets HIGH // digital output//  
Motorinp-1 pin gets LOW // digital output//  
delay(y1); }  
else if(val>x2 && val<x3) //Rain Sensor Values such that x3>x2//  
{ Motorinp -1 pin gets HIGH // digital output//  
Motorinp-2 pin gets LOW // digital output//  
delay(y2); //  
Motorinp-2 pin gets HIGH // digital output//  
Motorinp-1 pin gets LOW // digital output//  
delay(y2); }  
else if(val>x3 && valx3//  
{ Motorinp -1 pin gets HIGH // digital output//  
Motorinp-2 pin gets LOW // digital output//  
delay(y3);  
Motorinp-2 pin gets HIGH // digital output//  
Motorinp-1 pin gets LOW // digital output//

delay(y3); }  
else if(val<x0) //Rain Sensor Values such that x0< x1  
{ Motorinp-1 gets LOW // digital output//  
Motorinp-2 gets LOW // digital output// } }

Arduino Nano Interfacing with LED:

LED (+) Anode Pin -> Arduino Nano D8 Pin  
LED (-) Cathode Pin -> Arduino Nano GND Pin

Arduino Nano Interfacing with Rain Sensor:

Rain Sensor S Pin -> Arduino Nano A0 Pin  
Rain Sensor (-) GND Pin -> Arduino Nano GND Pin  
Rain Sensor (+) VCC Pin -> Arduino Nano D10 Pin

Arduino Nano Interfacing with Servo Motor (SG-90):

Servo Motor (SG-90) S Orange Color Pin -> Arduino Nano D9 Pin  
Servo Motor (SG-90) (-) Brown Color GND Pin -> Arduino Nano GND Pin  
Servo Motor (SG-90) (+) VCC Red Color Pin -> Arduino Nano 5V Pin

Arduino Nano Interfacing with HC-05 Bluetooth Module:

HC-05 Bluetooth Module RX Pin -> Arduino Nano TX1 Pin  
HC-05 Bluetooth Module TX Pin -> Arduino Nano Rx0 Pin  
HC-05 Bluetooth Module (-) GND Pin -> Arduino Nano GND Pin  
HC-05 Bluetooth Module (+) VCC Pin -> Arduino Nano 5V Pin

Arduino Nano Interfacing with l293D H-Bridge Motor Driver IC:-  
l293D 2nd Pin -> Arduino Nano D11 Pin  
l293D 7th Pin -> Arduino Nano D11 Pin