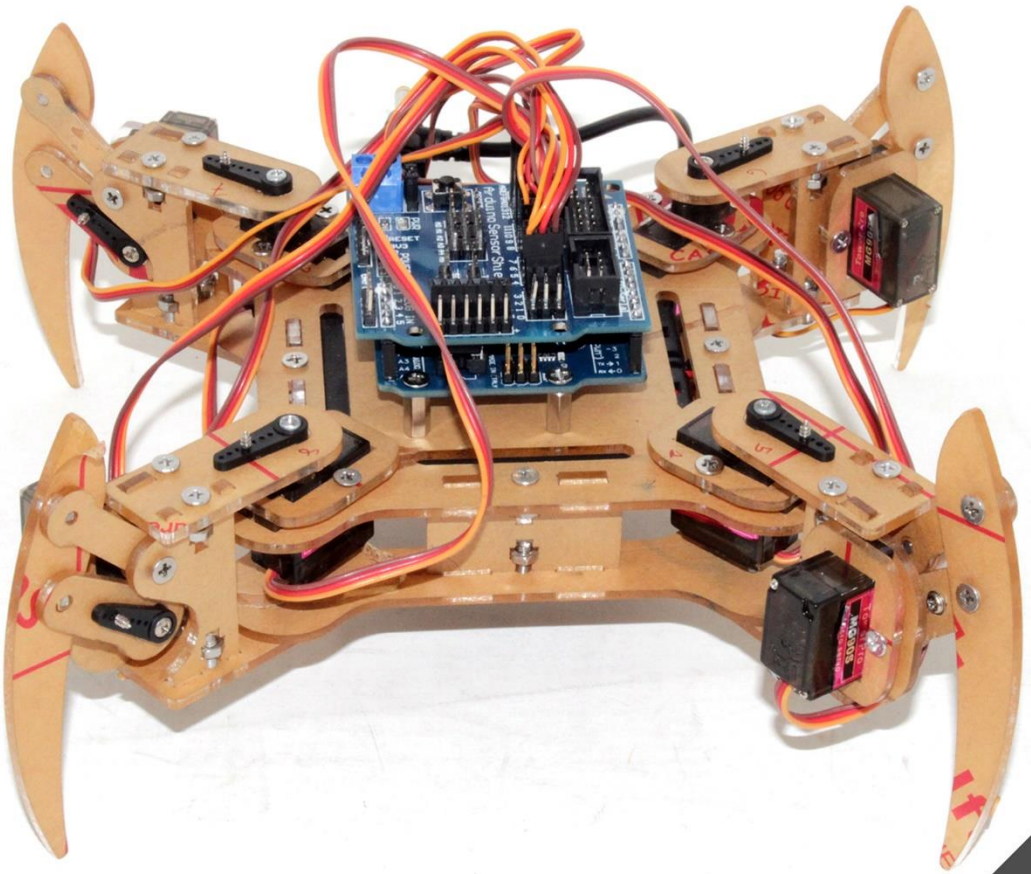
MePed
Quadpod Robotics



SUNROBOTICS

www.sunrobotics.in

SunRobotics Technologies
C-301, Sumel Business Park 6, Dudheshwar Road, Ahmedabad-380004, Gujarat-India
Ph no: 95588-27732 Email: support@sunrobotics.in

Preface

Sun Robotics is a technical service team of open source software and hardware. Dedicated to applying the Internet and the latest industrial technology in open source area, we strive to provide best hardware support and software service for general makers and electronic enthusiasts around the world. We aim to create infinite possibilities with sharing. No matter what field you are in, we can lead you into the electronic world and bring your ideas into reality.

This is a Professional Level kit for Arduino. Some common electronic components and sensors are included. Through the learning, you will get a better understanding of Arduino, and be able to make fascinating works based on Arduino.

Contents

Getting Started with Arduino

What is an Arduino?

Arduino is an open-source physical computing platform designed to make experimenting with electronics more fun and intuitive. Arduino has its own unique, simplified programming language, a vast support network, and thousands of potential uses, making it the perfect platform for both beginner and advanced DIY enthusiasts.

www.arduino.cc

A Computer for the Physical World:

The friendly blue board in your hand (or on your desk) is the Arduino. In some ways you could think of Arduino as the child of traditional desktop and laptop computers. At its roots, the Arduino is essentially a small portable computer. It is capable of taking inputs (such as the push of a button or a reading from a light sensor) and interpreting that information to control various outputs (like a blinking LED light or an electric motor). That's where the term "physical computing" is born - an Arduino is capable of taking the world of electronics and relating it to the physical world in a real and tangible way. Trust us - this will all make more sense soon.

Arduino UNO SMD R3

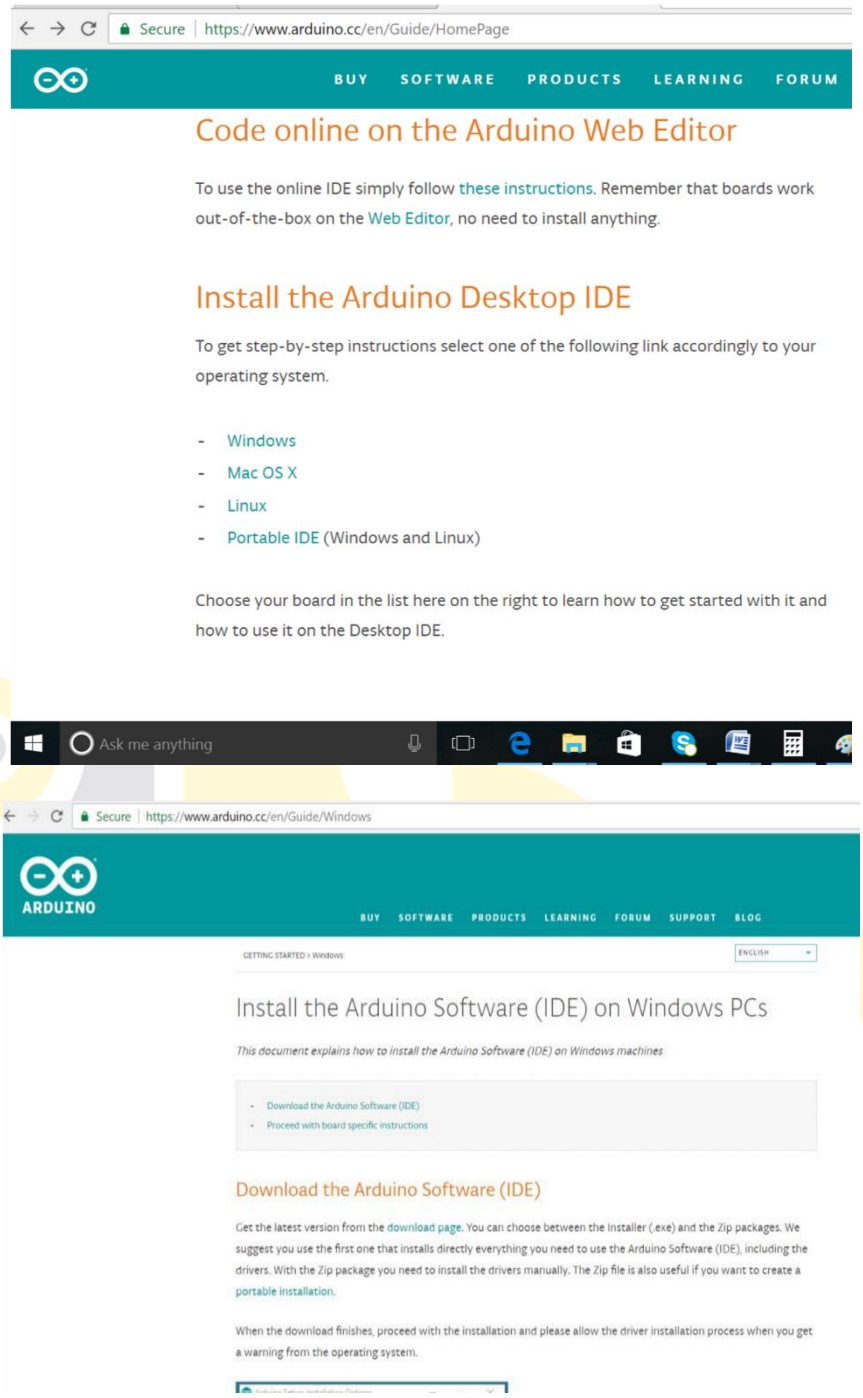
The Arduino Uno is one of several development boards based on the ATmega328. We like it mainly because of its extensive support network and its versatility. It has 14 digital input/output pins (6 of which can be PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Don’t worry, you’ll learn about all these later.

Installing Arduino IDE and Using Uno R3 board

STEP-1: Download the Arduino IDE (Integrated Development Environment)

Access the Internet: In order to get your Arduino up and running, you'll need to download some software first from www.arduino.cc (it's free!). This software, known as the Arduino IDE, will allow you to program the Arduino to do exactly what you want. It’s like a word processor for writing programs. With an internet-capable computer, open up your favorite browser and type in the following URL into the address bar:

www.arduino.cc/en/Main/Software



For different operating system platforms, the way of using Arduino IDE is different. Please refer to the following links: Windows User ： http://www.arduino.cc/en/Guide/Windows Mac OS User:http://www.arduino.cc/en/Guide/MacOSXLinuxUserhttp://playground.arduino.cc/Learning/Linux For more detailed information about Arduino IDE, please refer to the following link: **http://www.arduino.cc/en/Guide/HomePage**

STEP-2: Connect your Arduino Uno to your Computer:

Use the USB cable provided in the kit to connect the Arduino to one of your computer's USB inputs.

STEP-3: Install Drivers

Depending on your computer's operating system, you will need to follow specific instructions. Please go to the URLs below for specific instructions on how to install the drivers onto your Arduino Uno.

Windows Installation Process: Go to the web address below to access the instructions for installations on a Windows-based computer.

http://arduino.cc/en/Guide/Windows

Macintosh OS X Installation Process: Macs do not require you to install drivers. Enter the following URL if you have questions. Otherwise proceed to next page.

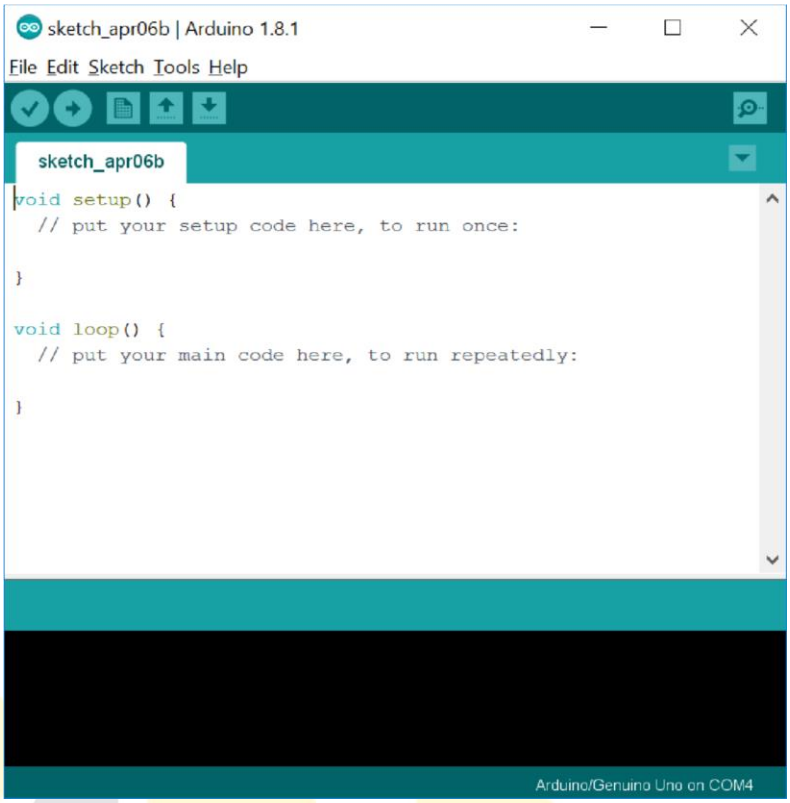
http://arduino.cc/en/Guide/MacOSX

Linux: 32 bit / 64 bit, Installation Process Go to the web address below to access the instructions for installations on a Linux-based computer.

http://www.arduino.cc/playground/Learning/Linux

STEP-4: Open the Arduino IDE

Open the Arduino IDE software on your computer. Poke around and get to know the interface. We aren't going to code right away, this is just an introduction. The step is to set your IDE to identify your Arduino Uno.



GUI (Graphical User Interface)

Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for Details .Note: If you are using an external programmer with your board, you can hold Down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer" New Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within The current window overwriting its content. Note: due to a bug in Java, this menu Doesn't scroll; if you need to open a sketch late in the list, use the File | Sketch book Menu instead.

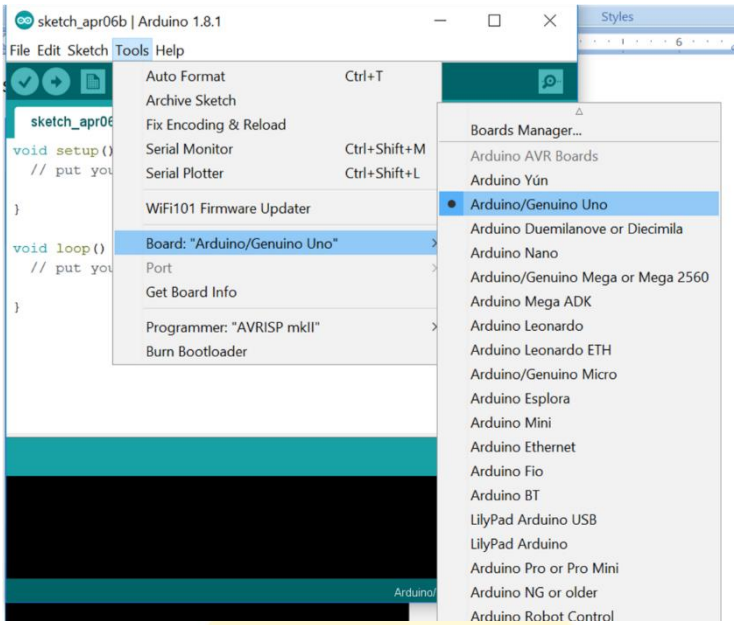
Save

Saves your sketch.

Serial Monitor

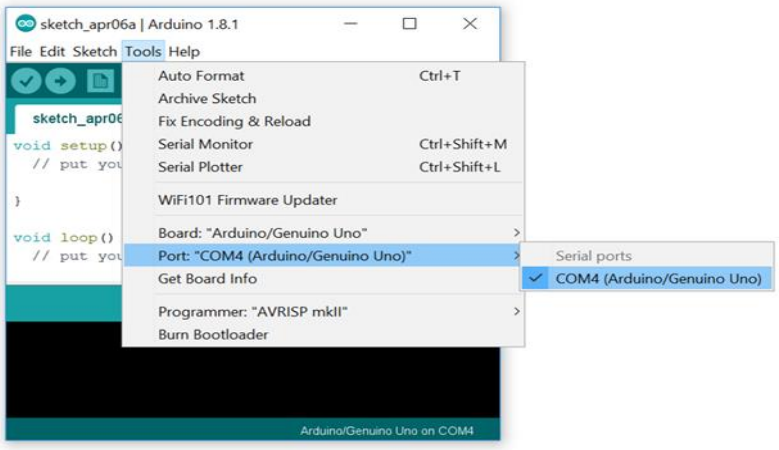
Opens the serial monitor.

STEP-5: Select your board: Arduino Uno



STEP-6: Select your Serial Device

Windows: Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be com3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.



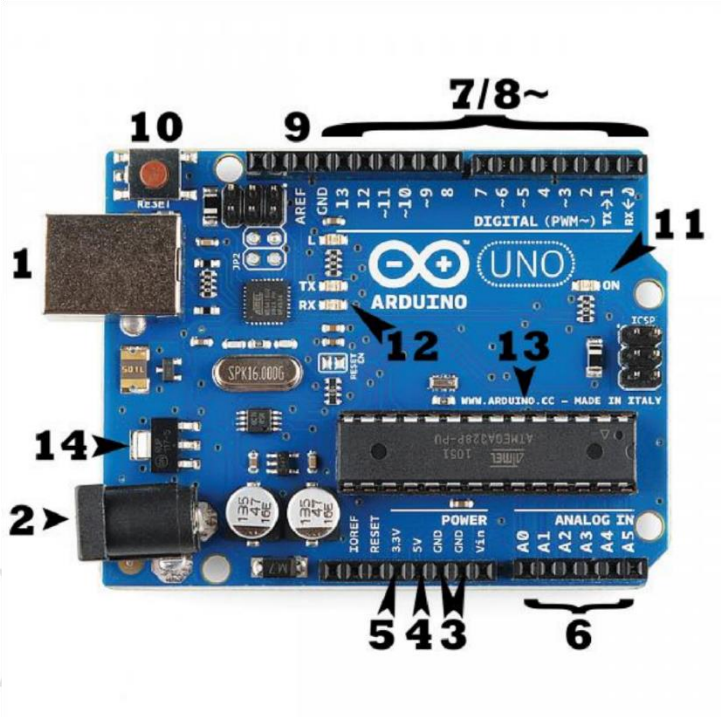
Mac OS: Select the serial device of the Arduino board from the Tools > Serial Port menu. On the Mac, this should be something with /dev/tty.usbmodem (for the Uno or Mega 2560) or /dev/tty.usbserial (for older boards) in it.

Linux: http://playground.arduino.cc/Learning/Linux

About Arduino Uno R3 board

What's on the board?

There are many varieties of Arduino boards that can be used for different purposes. Some boards look a bit different from the one below, but most Arduino have the majority of these components in common:



Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2).** .

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts. Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3):** Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5):** As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6):** The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

Main IC

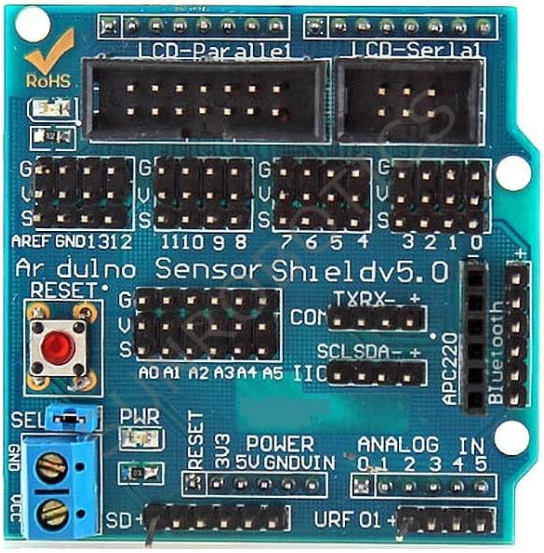
The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the AT mega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

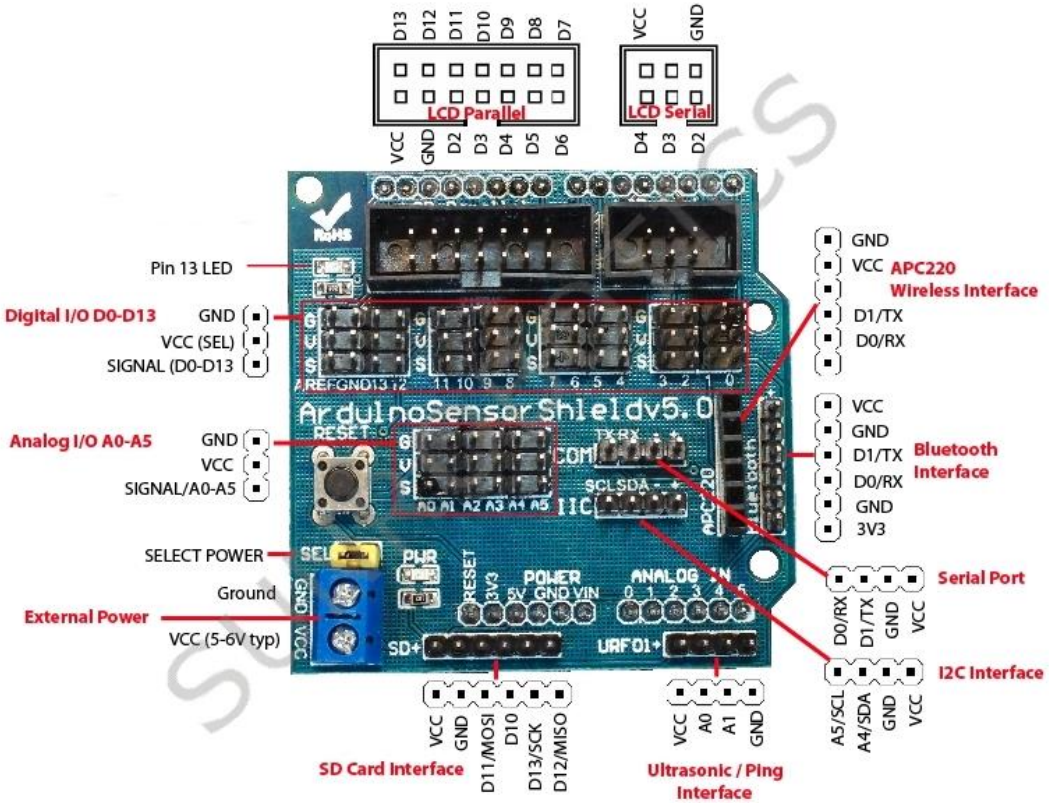
Voltage Regulator

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the

Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

About Arduino Sensor Shield Module





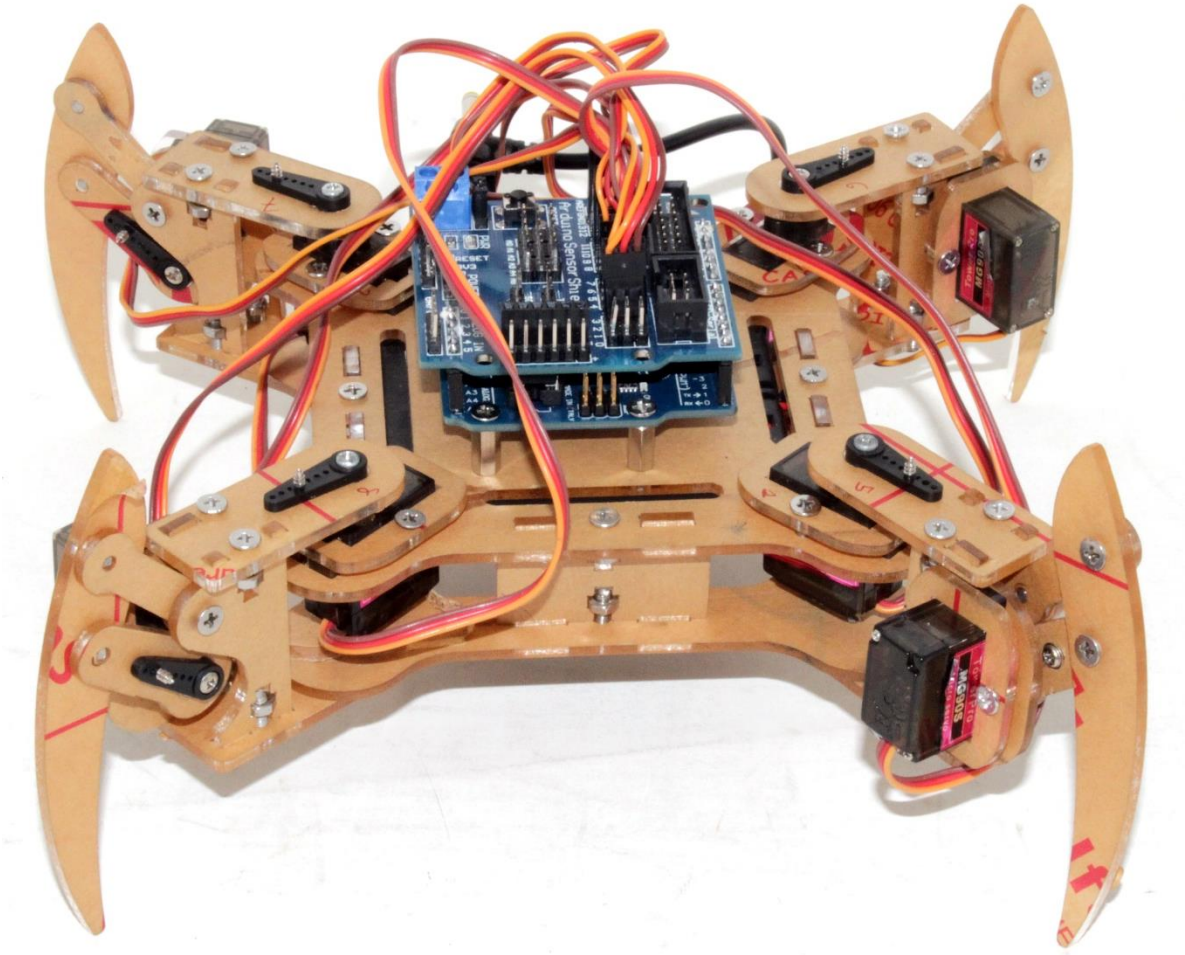
This shield board plugs on top of an Arduino and makes it easy to connect to all the Digital Inputs and Outputs and the Analog Inputs. It also makes it easy to connect to relays, LEDs etc.

The 5X Version has the ability to optionally separate the 5V supply going to all the 3-pin connectors and feed them from a separate external power supply. This is ideal for running lots of Servos or other power devices.

3 pins are available for each port, Voltage, Ground and Signal. There is also a Reset button.

Connect all module with MePed Chassis

Refer MePed assembly manual to connect all module with MePed chassis.



Lesson 1- Controlling a Servo Using Arduino

Overview:

In this lesson, we will introduce a new electronic device (Servo) to you, and tell you how to control it with the Arduino UNO.

Components:

- 1 x Arduino UNO
- 1 x USB Cable
- 1 x Servo
- Arduino Sensor Shield
- Several jumper wires

Principle:

Driving the servos with Arduino sensor shield. the arduino shield actually breaks out Arduino's 16bit PWM output pins #9 & #10 to the edge of the shield with two 3-pin headers.
Power for the Servos comes from the Arduino's on-board 5V regulator, if you drive multiple servos you need EXT_PWR supply which on arduino sensor shield board.

1. Servo motor

The servo motor has three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino board. The ground wire is typically black or brown and should be connected to a ground pin on the Arduino board. Usually the signal pin is yellow, orange or white, and should be connected to a digital pin on the Arduino board. Note that the servo motor draws a considerable amount of power, if you need to drive more than one or two servos, you'll probably need to power them with an extra supply (i.e. not the +5V pin on your Arduino). Be sure to connect the grounds of the Arduino and external power supply together.

2. Servo library

This library allows an Arduino board to control RC (hobby) servo motors. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds.

3. Key functions:

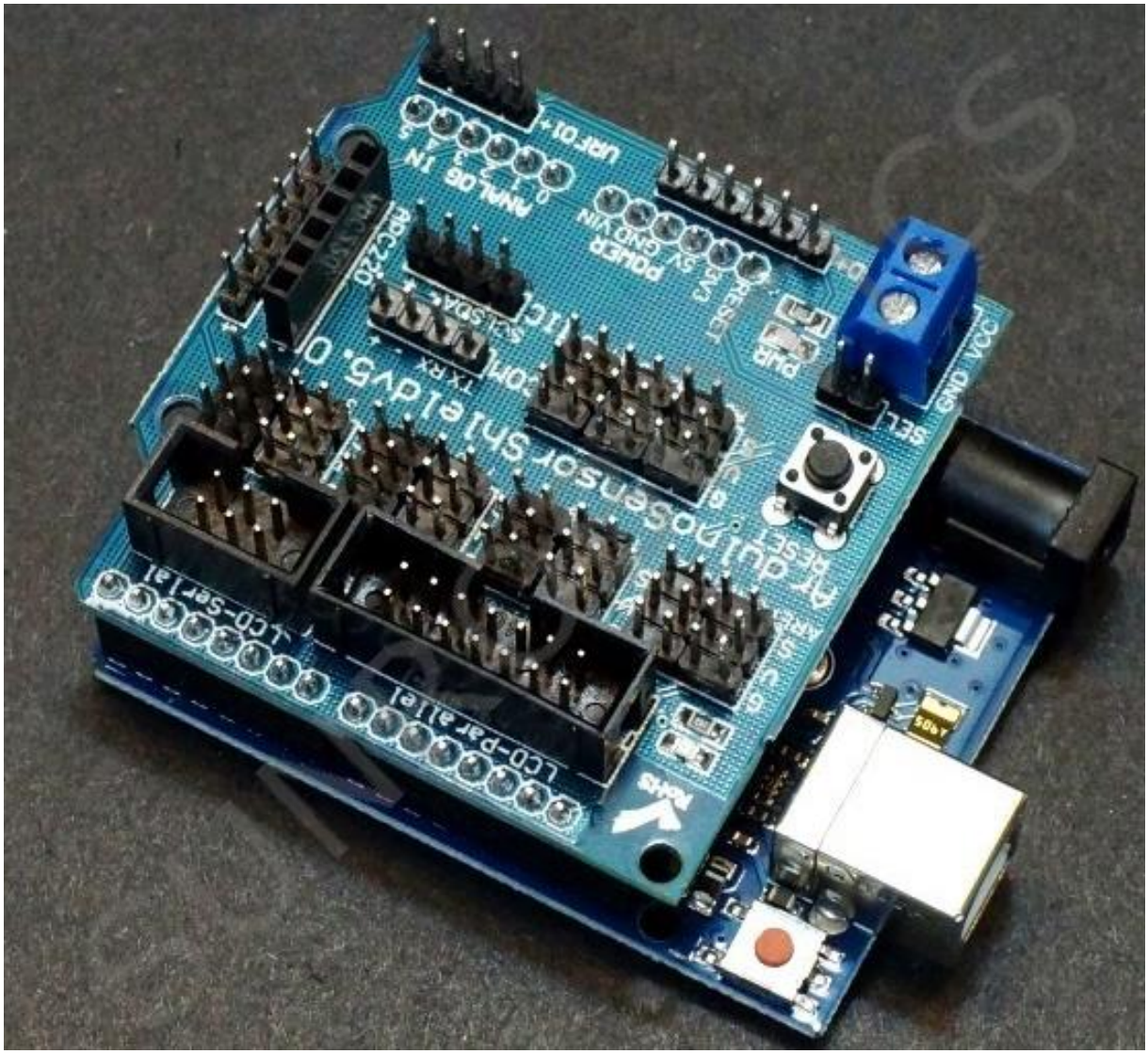
attach()

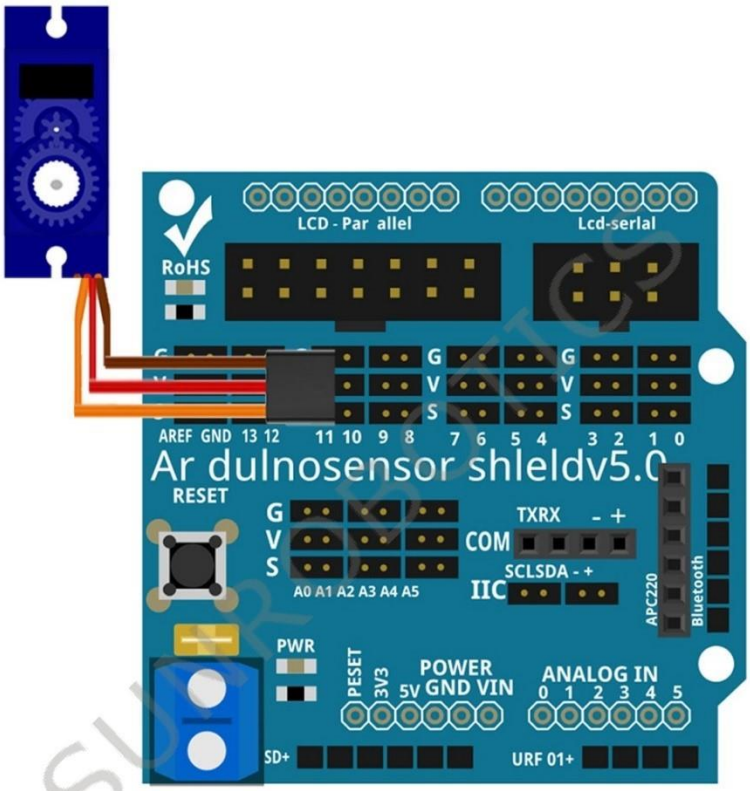
Attach the Servo variable to a pin. Note that in Arduino 0016 and earlier, the Servo library supports only servos on only two pins: 9 and 10.
Syntax:
servo.attach(pin)
servo.attach(pin, min, max)
Parameters:
servo: a variable of type Servo
pin: the number of the pin that the servo is attached to

min (optional): the pulse width, in microseconds, corresponding to the minimum (0-degree) angle on the servo (defaults to 544)
max (optional): the pulse width, in microseconds, corresponding to the maximum (180-degree) angle on the servo (defaults to 2400)

Procedure:

Step 1: Build the circuit.





Arduino sensor shield V5.0	Servo motor
D11	S
5V	V
GND	G

Step 2: Program:

> **NOTE:**
> Before you upload the program, you need to install the Servo.zip library. Otherwise the program will throw an error message.

Step 3: Compile the program and upload to Arduino UNO board

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int pos = 0;  // variable to store the servo position
void setup()
{
  // attaches the servo on pin 10 to the servo object
  myservo.attach(10);
}
void loop()
{
  myservo.write(15);//goes to 15 degrees
```

```
  delay(1000);//wait for a second

  myservo.write(30);//goes to 30 degrees
  delay(1000);//wait for a second.33

  myservo.write(45);//goes to 45 degrees
  delay(1000);//wait for a second.33

  myservo.write(60);//goes to 60 degrees
  delay(1000);//wait for a second.33

  myservo.write(75);//goes to 75 degrees
  delay(1000);//wait for a second.33

  myservo.write(90);//goes to 90 degrees
  delay(1000);//wait for a second

  myservo.write(75);//back to 75 degrees
  delay(1000);//wait for a second.33

  myservo.write(60);//back to 60 degrees
  delay(1000);//wait for a second.33

  myservo.write(45);//back to 45 degrees
  delay(1000);//wait for a second.33

  myservo.write(30);//back to 30 degrees
  delay(1000);//wait for a second.33

  myservo.write(15);//back to 15 degrees
  delay(1000);//wait for a second

  myservo.write(0);//back to 0 degrees
  delay(1000);//wait for a second

  for(int num=0;num<=180;num++)
  {
    myservo.write(num);//back to 'num' degrees(0 to 180)
    delay(10);//control servo speed
  }
  for(int num=180;num>=0;num--)
  {
    myservo.write(num);//back to 'num' degrees(180 to 0)
    delay(10);//control servo speed
  }
}
```

Now, you should see the servo rotate from 0 to 180 degrees, and then do it in the opposite direction.

Lesson 2– HC05 UART Communication with arduino

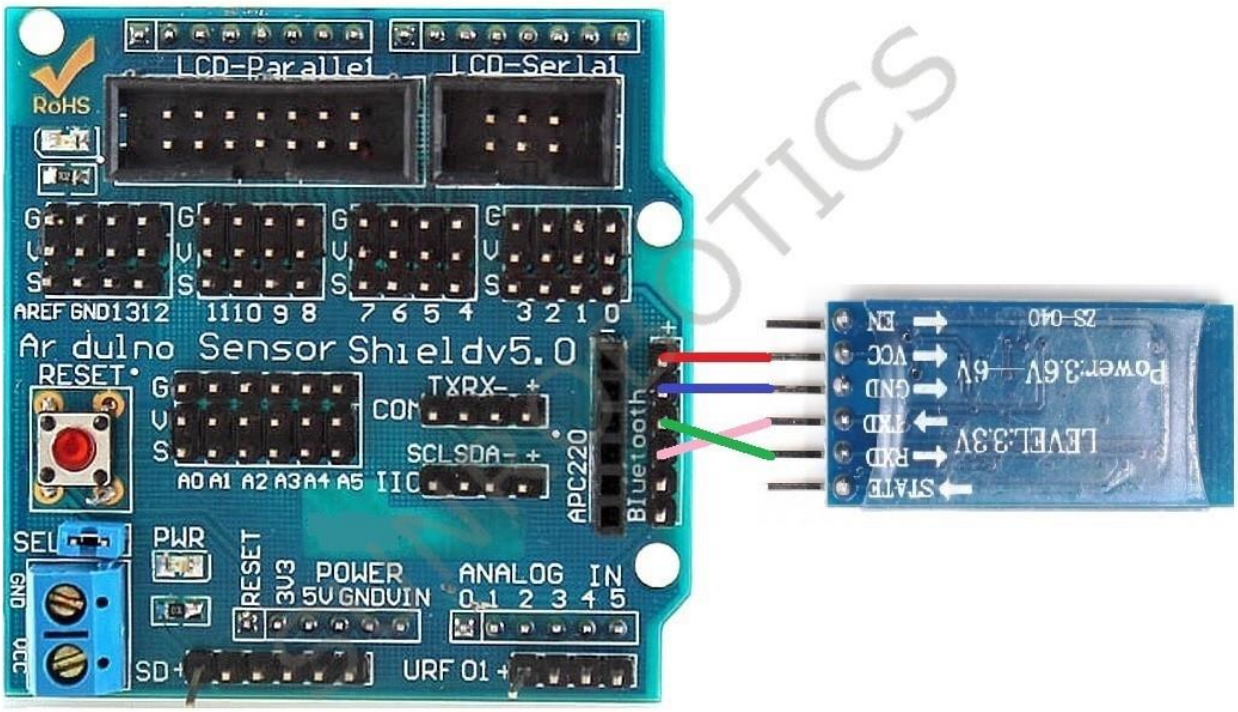
Overview:
In this tutorial we will study about the arduino using Bluetooth module HC-05.
Components:

- 1 x Arduino Uno
- 1 x USB Cable
- 1 x Bluetooth Module HC-05
- 1 x Arduino Sensor shield Module
- Jumper wires

Principle: Arduino can communicate with other devices via Bluetooth using the module HC-05 (master/slave). It enables the Arduino to connect and exchange data with other devices such as Smartphone, computer or other microcontrollers. Bluetooth communication can be used to control a robot remotely, Display and store data on your computer or on your smartphone, for instance.

Schematic:
Vcc – Arduino 5v
GND – Arduino GND
TXD – Arduino Pin D0

Procedure:
Step 1: Build the circuit



NOTE:
Before you upload the program, you need to Bluetooth TX and RX pin disconnect from Arduino Sensor shield board. Otherwise the program will throw an error message. After Program upload done connect TX, RX pin with board.

Step 2: Program: Open /Copy the code from the “CODE” Folder

```
/*
  HC05 - Bluetooth AT-Command mode
*/
#include "SoftwareSerial.h"
SoftwareSerial MyBlue(0, 1); // RX | TX
char c = ' ';
void setup()
{
  Serial.begin(9600);
  MyBlue.begin(9600);  //Baud Rate for AT-command Mode.
  Serial.println("***AT commands mode***");
}
void loop()
{
  //from bluetooth to Terminal.
  if (MyBlue.available())
  {
    c = MyBlue.read();
    Serial.write(c);
  }
  //from termial to bluetooth
  if (Serial.available())
  {
    c = Serial.read();
    MyBlue.write(c);
  }
}
```

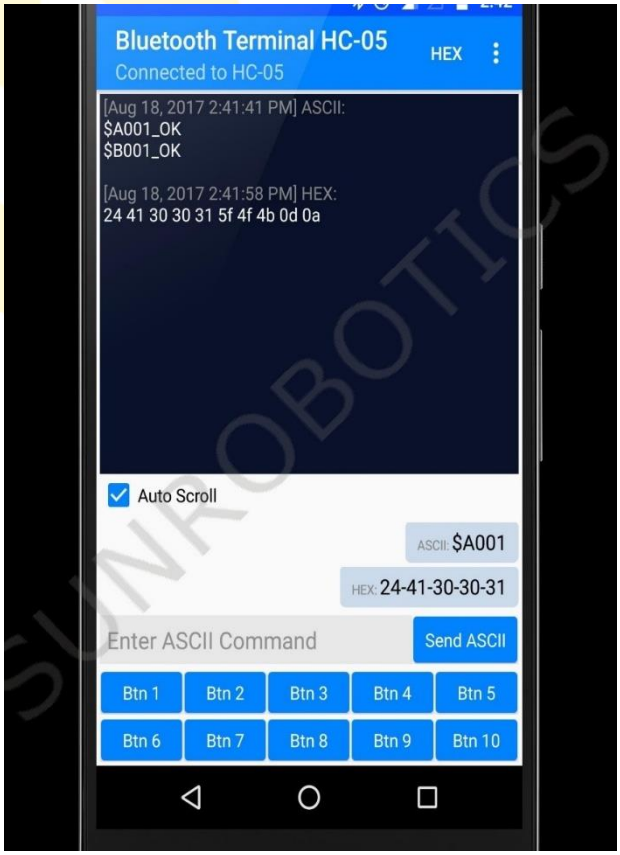
Step 3: Compile the program and upload to Arduino UNO board.

Step 4: “Bluetooth Terminal HC-05” free App Download In your phone from Play Store. Btn command set anything “B” ,“A” etc. then when app button press command send to arduino terminal via Bluetooth.

i. Open application

ii. Scan devices from this pair and connect

iii. Set btn Command and then press button



iv. this Command output also see in arduino serial monitor

AT Commands

In general, typing the command **AT+<command>?** will prompt the saved parameter (ex: AT+PSWD? will display the module PIN code). If you enter **AT+<command>=<Param>**, you can set the parameter value(ex: AT+PWSD=0000 to modify the PIN code to 0000).
Here are some of the AT commands:
To test communication,enter **AT** in the serial monitor of the Arduino IDE. If everything is setup correctly it should display OK.
To modify the module name, enter **AT+NAME=<Param>**. The module should answer OK (Defaut HC-05, Ex: To change the name to BTM1 enter AT+NAME=BTM1).
To modify the PIN code, enter **AT+PSWD=<Param>** . The module should answer OK(Default 1234 Ex: To change the PIN to 0000 enter AT+PSWD=0000).
AT+ROLE=<Param> to midy the role of the module as slave or master (Default 0, Ex: to change the role as master enter AT+ROLE=1, as slave enter AT+ROLE=0).
To modify the baudrate, enter **AT+UART=<Param1>,<Param2>,<Param3>** with Param1, 2 and 3 serial communication parameters: baudrate, stop bit and parity bit respectively (By default,set to 9600,0,0. Ex: to modify the baudrate to 115200 enter AT+UART=115200,0,0).
Other AT commands exist for the Bluetooth module HC-05 that you can find following the link.

Slave Configuration
To set the module as a slave, you can change the name as AT+NAME=HC05-Slave and choose the communication parameters and the PIN code that you want. You'll need to make sure that master and slave as the same communication parameters.
AT returns OK
AT+NAME=HC05-Slave
AT+UART=9600,0,0
AT+ROLE=0
Enter AT+ADDR to obtain the module address (ex: +ADDR:98d3:32:21450e)

Master Configuration
To set the module as master, you need to change the role and set the same communication parameter as the slave module.
AT returns OK
AT+NAME=HC05-Master
AT+UART=9600,0,0
AT+ROLE=1
The slave module address must be enter in the master module to allows it to appair: AT+BIND=98d3,32,21450e (replace dots ":" by coma ",")

Lesson- 3 Set 0 Position of servo motor before connect chassis

Overview:

In this tutorial we will study set 0 position of servo motors.

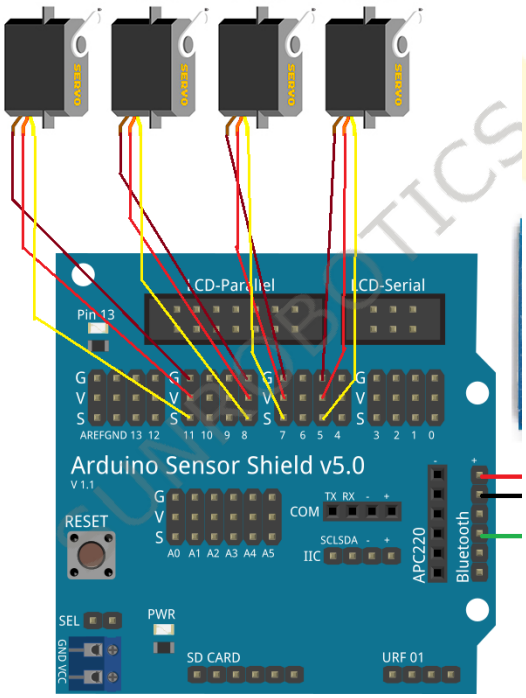
Components:

- 1 x Arduino Uno
- 1 x USB Cable
- 1 x Arduino Sensor shield Module
- 8 x servo motors(5v based)
- Male to female Jumper wires

Schematic connections:

- Servo Motors with arduino sensor shield board
  Front Left Pivot Servo into S1 – Arduino sensor shield pin number D1
  Back Left Pivot Servo into S3 – Arduino sensor shield pin number D3
  Back Right Pivot Servo into S5 – Arduino sensor shield pin number D5
  Front Right Pivot Servo into S7 – Arduino sensor shield pin number D7
  Front Left Lift Servo into S2 – Arduino sensor shield pin number D2
  Back Left Lift Servo into S4 – Arduino sensor shield pin number D4
  Back Right Lift Servo into S6 – Arduino sensor shield pin number D6
  Front Right Lift Servo into S8 – Arduino sensor shield pin number D8
  All red pin is vcc which can connect with 5v supply
  All brown pin is gnd which can connect with ground
  All orange pin is output

Step 1: Build the circuit



Step 2: Program: Open /Copy the code from the “CODE” Folder

```
#include <Servo.h>

int servoPin1 = 7;
int servoPin2 = 13;
int servoPin3 = 5;
int servoPin4 = 12;
int servoPin5 = 3;
int servoPin6 = 10;
int servoPin7 = 2;
int servoPin8 = 8;
int servoPinu = 11;
unsigned char i;
/*Front Left Pivot Servo into S1
  Front Left Lift Servo into S2
  Back Left Pivot Servo into S3
  Back Left Lift Servo into S4
  Back Right Pivot Servo into S5
  Back Right Lift Servo into S6
  Front Right Pivot Servo into S7
  Front Right Lift Servo into S8*/
Servo Servo1, Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servou;
void setup() {
  Servo1.attach(servoPin1); //Front Left Pivot Servo into S1
  Servo2.attach(servoPin3); //Back Left Pivot Servo into S3
  Servo3.attach(servoPin5); // Back Right Pivot Servo into S5
  Servo4.attach(servoPin7); //Front Right Pivot Servo into S7
  Servo5.attach(servoPin2); //Front Left Lift Servo into S2
  Servo6.attach(servoPin4); //Back Left Lift Servo into S4
  Servo7.attach(servoPin6); //Back Right Lift Servo into S6
  Servo8.attach(servoPin8); //Front Right Lift Servo into S8

  Servo1.write(0);
  Servo2.write(0);
  Servo3.write(0);
  Servo4.write(0);
  Servo5.write(0);
  Servo6.write(0);
  Servo7.write(0);
  Servo8.write(0);
}
void loop()
{
}
```

Step 3: Compile the program and upload to Arduino UNO board.

Lesson-4 MePed run automatic

Overview:

In this tutorial Run the MePed robot automatic in any one direction. Here in this code MePed robot run in forward direction.

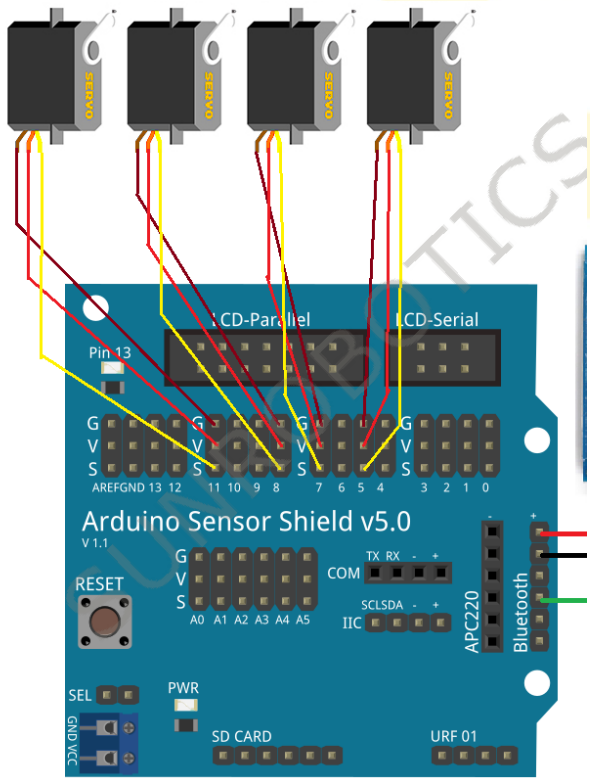
Components:

- 1 x Arduino Uno
- 1 x USB Cable
- 1 x Arduino Sensor shield Module
- 8 x servo motors(5v based)
- Male to female Jumper wires

Schematic connections:

- Servo Motors with arduino sensor shield board
  Front Left Pivot Servo into S1 – Arduino sensor shield pin number D1
  Back Left Pivot Servo into S3 – Arduino sensor shield pin number D3
  Back Right Pivot Servo into S5 – Arduino sensor shield pin number D5
  Front Right Pivot Servo into S7 – Arduino sensor shield pin number D7
  Front Left Lift Servo into S2 – Arduino sensor shield pin number D2
  Back Left Lift Servo into S4 – Arduino sensor shield pin number D4
  Back Right Lift Servo into S6 – Arduino sensor shield pin number D6
  Front Right Lift Servo into S8 – Arduino sensor shield pin number D8
  All red pin is vcc which can connect with 5v supply
  All brown pin is gnd which can connect with ground
  All orange pin is output

Step 1: Build the circuit



Step 2: Program: Open /Copy the code from the “CODE” Folder

```
// Include the Servo library
#include <Servo.h>

int servoPin1 = 7;
int servoPin2 = 13;
int servoPin3 = 5;
int servoPin4 = 12;
int servoPin5 = 3;
int servoPin6 = 10;
int servoPin7 = 2;
int servoPin8 = 8;
int servoPinu = 11;
unsigned char i;
/*Front Left Pivot Servo into S1
  Front Left Lift Servo into S2
  Back Left Pivot Servo into S3
  Back Left Lift Servo into S4
  Back Right Pivot Servo into S5
  Back Right Lift Servo into S6
  Front Right Pivot Servo into S7
  Front Right Lift Servo into S8*/
Servo Servo1, Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servou;
void setup() {
  Servo1.attach(servoPin1); //Front Left Pivot Servo into S1
  Servo2.attach(servoPin3); //Back Left Pivot Servo into S3
  Servo3.attach(servoPin5); // Back Right Pivot Servo into S5
  Servo4.attach(servoPin7); //Front Right Pivot Servo into S7
  Servo5.attach(servoPin2); //Front Left Lift Servo into S2
  Servo6.attach(servoPin4); //Back Left Lift Servo into S4
  Servo7.attach(servoPin6); //Back Right Lift Servo into S6
  Servo8.attach(servoPin8); //Front Right Lift Servo into S8

  Servo1.write(90);
  Servo2.write(90);
  Servo3.write(90);
  Servo4.write(90);
  Servo5.write(90);
  Servo6.write(90);
  Servo7.write(90);
  Servo8.write(90);
}
void loop() {

  /*
    /////// STAND UP - DOWN START /////////////////
    Servo5.write(130);//leg DOWN - Front Left Lift Servo into
```

```
  delay(100);
  Servo6.write(130);// leg DOWN - Back Right Lift Servo
  delay(100);
  Servo7.write(130); // leg DOWN - Back Right Lift Servo
  delay(100);
  Servo8.write(130); // leg DOWN - Front Right Lift Servo
  delay(6000);

  Servo5.write(50);//leg UP - Front Left Lift Servo into
  delay(100);
  Servo6.write(50); // leg UP - Back Right Lift Servo
  delay(100);
  Servo7.write(50); // leg UP - Back Right Lift Servo
  delay(100);
  Servo8.write(50); // leg UP - Front Right Lift Servo
  delay(6000);
  ////////// STAND UP - DOWN END ////////////////
 */


//////// RUN FORWARD START ////////////////////////////////

///// LEFT SIDE LEGS
Servo6.write(50); // leg UP - Back Right Lift Servo
delay(100);
Servo2.write(30); //right direction - Back Left Pivot Servo
delay(100);
Servo1.write(130); //left direction - Front Left Pivot Servo
delay(100);
Servo6.write(100); // leg Down -Back Right Lift Servo
delay(100);
Servo5.write(50); //leg UP - Front Left Lift Servo
delay(100);
Servo1.write(70); // right direction - Front Left Pivot Servo
delay(100);
Servo2.write(90); // left direction - -Back Left Pivot Servo
delay(100);
Servo5.write(100); // leg Down -Front Left Lift Servo
delay(100);

///////RIGHT SIDE LEGS
Servo7.write(50);// leg UP - Back Right Lift Servo
delay(100);
Servo3.write(150); // left direction - Back Right Pivot Servo
delay(100);
Servo4.write(30);// right direction -Front Right Pivot Servo
delay(100);
Servo7.write(100);//leg Down - Back Right Lift Servo
delay(100);
```

```
Servo8.write(50);// leg UP - Front Right Lift Servo
delay(100);
Servo4.write(90);// left direction -Front Right Pivot Servo
delay(100);
Servo3.write(85); // right direction -Back Right Pivot Servo
delay(100);
Servo8.write(100);//leg Down - Front Right Lift Servo
delay(100);

////////// /////// RUN FORWARD END /////////////////


/*
 //////// RUN REVERSE START ////////////////////////////////

 ///////RIGHT SIDE LEGS
 Servo8.write(50); // leg UP - Front Right Lift Servo
 delay(100);
 Servo4.write(30); //right direction - Front Right Pivot Servo
 delay(100);
 Servo3.write(130); //left direction - Back Right Pivot Servo
 delay(100);
 Servo8.write(100); // leg Down -Front Right Lift Servo
 delay(100);
 Servo7.write(50); //leg UP - Back Right Lift Servo
 delay(100);
 Servo3.write(70); // right direction - Back Right Pivot Servo
 delay(100);
 Servo4.write(90); // left direction - Front Right Pivot Servo
 delay(100);
 Servo7.write(100); // leg Down - Back Right Lift Servo
 delay(100);

 ///// LEFT SIDE LEGS
 Servo5.write(50);// leg UP - Front Left Lift Servo
 delay(100);
 Servo1.write(150); // left direction - Front Left Pivot Servo
 delay(100);
 Servo2.write(30);// right direction -Back Left Pivot Servo
 delay(100);
 Servo5.write(100);//leg Down - Front Left Lift Servo
 delay(100);
 Servo6.write(50);// leg UP - Back Right Lift Servo
 delay(100);
 Servo2.write(90);// left direction -Back Left Pivot Servo
 delay(100);
 Servo1.write(85); // right direction -Front Left Pivot Servo
 delay(100);
 Servo6.write(100);//leg Down - Back Right Lift Servo
```

```
  delay(100);

  ////////// /////// RUN REVERSE END /////////////////
 */

/*
  //////// RUN LEFT START ////////////////////////////////

  Servo5.write(50);// leg UP - Front Left Lift Servo
  delay(100);
  Servo1.write(160); // left direction - Front Left Pivot Servo
  delay(100);
  Servo2.write(90); // middle direction - Back Left Pivot Servo
  delay(100);
  Servo5.write(100);//leg Down - Front Left Lift Servo
  delay(100);

  Servo8.write(50);// leg UP - Front Right Lift Servo
  delay(100);
  Servo4.write(160); // left direction - Front Right Pivot Servo
  delay(100);
  Servo1.write(90); // middle direction - Front Left Pivot Servo
  delay(100);
  Servo8.write(100);//leg Down - Front Right Lift Servo
  delay(100);

  Servo7.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo3.write(160); // left direction - Back Right Pivot Servo
  delay(100);
  Servo4.write(90); // middle direction - Front Right Pivot Servo
  delay(100);
  Servo7.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  Servo6.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo2.write(160); // left direction - Back Left Pivot Servo
  delay(100);
  Servo3.write(90); // middle direction - Back Right Pivot Servo
  delay(100);
  Servo6.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  ////////// /////// RUN LEFT END /////////////////
 */
/*
   //////// RUN RIGHT START ////////////////////////////////
```

```
  Servo5.write(50);// leg UP - Front Left Lift Servo
  delay(100);
  Servo1.write(20); // left direction - Front Left Pivot Servo
  delay(100);
  Servo2.write(90); // middle direction - Back Left Pivot Servo
  delay(100);
  Servo5.write(100);//leg Down - Front Left Lift Servo
  delay(100);

  Servo8.write(50);// leg UP - Front Right Lift Servo
  delay(100);
  Servo4.write(20); // left direction - Front Right Pivot Servo
  delay(100);
  Servo1.write(90); // middle direction - Front Left Pivot Servo
  delay(100);
  Servo8.write(100);//leg Down - Front Right Lift Servo
  delay(100);

  Servo7.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo3.write(20); // left direction - Back Right Pivot Servo
  delay(100);
  Servo4.write(90); // middle direction - Front Right Pivot Servo
  delay(100);
  Servo7.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  Servo6.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo2.write(20); // left direction - Back Left Pivot Servo
  delay(100);
  Servo3.write(90); // middle direction - Back Right Pivot Servo
  delay(100);
  Servo6.write(100);//leg Down - Back Right Lift Servo
  delay(100);
  ////////// /////// RUN RIGHT END ////////////////
 */
}
```

Step 3: Compile the program and upload to Arduino UNO board.

Lesson-5 MePed Robo Control Using Servo and Android app

Overview:
In this tutorial control the MePed robot using bluetooth android app.

Components:
- 1 x Arduino Uno
- 1 x USB Cable
- 1 x Arduino Sensor shield Module
- 8 x servo motors(5v based)
- Male to female Jumper wires
- 1 x bluetooth HC-05

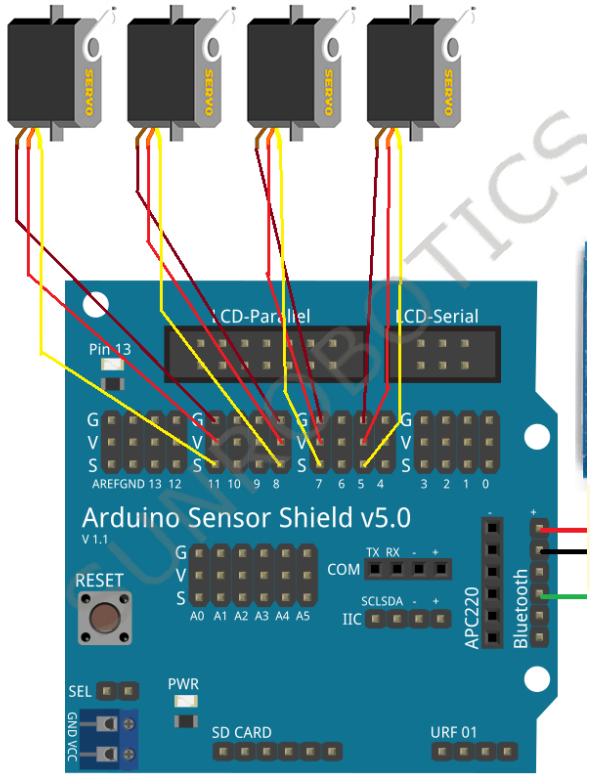
Schematic connections:
- Servo Motors with arduino sensor shield board
  Front Left Pivot Servo into S1 – Arduino sensor shield pin number D1
  Back Left Pivot Servo into S3 – Arduino sensor shield pin number D3
  Back Right Pivot Servo into S5 – Arduino sensor shield pin number D5
  Front Right Pivot Servo into S7 – Arduino sensor shield pin number D7
  Front Left Lift Servo into S2 – Arduino sensor shield pin number D2
  Back Left Lift Servo into S4 – Arduino sensor shield pin number D4
  Back Right Lift Servo into S6 – Arduino sensor shield pin number D6
  Front Right Lift Servo into S8 – Arduino sensor shield pin number D8
  All red pin is vcc which can connect with 5v supply
  All brown pin is gnd which can connect with ground
  All orange pin is output

- Bluetooth module connect with arduino sensor shield board
  TXD Pin connect with D0
  VCC Pin connect with 5v
  GND pin connect with ground

Principle: For the android communication with our Bluetooth module I've used the "Meped Robot control application"android app. Then you just need to connect your smartphone with the Bluetooth module. We can see in application ↑ ↓ ← → arrow and Lock menu.

Working:-
- when Autonom mode on –system work in continuity mode. i.e. when up arrow press one time system will lock this arrow position and robot car continue move in forward direction.
- when Autonom off- system work when key continues press. When release robot car will be stop.
- In lAutonom mode on condition app send command to arduino using hc-05 (when ↑ press send "F", when ↓ press send "B", when ← press send "L", when → press send "R")
- In Autonom off condition app send command to arduino using hc-05 (when ↑ press send "F" after release send "S", when ↓ press send "B" after release send "S", when ← press send "L" after release send "S", when → press send "R" after release send "S")
- Also SITDOWN – send command "D"
- Also STANDUP – send command "U"

Step 1: Build the circuit



Step 2: Program: Open /Copy the code from the “CODE” Folder

```
#include "SoftwareSerial.h"
#include <Servo.h>

int servoPin1 = 7;
int servoPin2 = 13;
int servoPin3 = 5;
int servoPin4 = 12;
int servoPin5 = 3;
int servoPin6 = 10;
int servoPin7 = 2;
int servoPin8 = 8;
int servoPinu = 11;
unsigned char i;
bool forward_flag = false;
bool reverse_flag = false;
bool left_flag = false;
bool right_flag = false;
bool standup_flag = false;
bool sitdown_flag = false;
bool stop_flag = false;
int state;
/*Front Left Pivot Servo into S1
```

```
  Front Left Lift Servo into S2
  Back Left Pivot Servo into S3
  Back Left Lift Servo into S4
  Back Right Pivot Servo into S5
  Back Right Lift Servo into S6
  Front Right Pivot Servo into S7
  Front Right Lift Servo into S8*/
SoftwareSerial MyBlue(0, 1); // RX | TX
Servo Servo1, Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servou;
void setup() {
  Serial.begin(9600);
  MyBlue.begin(9600);  //Baud Rate for AT-command Mode.
  Servo1.attach(servoPin1); //Front Left Pivot Servo into S1
  Servo2.attach(servoPin3); //Back Left Pivot Servo into S3
  Servo3.attach(servoPin5); // Back Right Pivot Servo into S5
  Servo4.attach(servoPin7); //Front Right Pivot Servo into S7
  Servo5.attach(servoPin2); //Front Left Lift Servo into S2
  Servo6.attach(servoPin4); //Back Left Lift Servo into S4
  Servo7.attach(servoPin6); //Back Right Lift Servo into S6
  Servo8.attach(servoPin8); //Front Right Lift Servo into S8

  Servo1.write(90);
  Servo2.write(90);
  Servo3.write(90);
  Servo4.write(90);
  Servo5.write(90);
  Servo6.write(90);
  Servo7.write(90);
  Servo8.write(90);
}
void loop() {
  if (MyBlue.available() > 0)
  {
    state = MyBlue.read();
    Serial.println(state);
    if (state == 'F')
    {
      forward_flag = true;
      reverse_flag = false;
      left_flag = false;
      right_flag = false;
      standup_flag = false;
      sitdown_flag = false;
    }
    if (state == 'B')
    {
      reverse_flag = true;
      forward_flag = false;
      left_flag = false;
```

```
  right_flag = false;
  standup_flag = false;
  sitdown_flag = false;
 }
 if (state == 'L')
 {
  left_flag = true;
  forward_flag = false;
  reverse_flag = false;
  right_flag = false;
  standup_flag = false;
  sitdown_flag = false;
 }
 if (state == 'R')
 {
  right_flag = true;
  forward_flag = false;
  reverse_flag = false;
  left_flag = false;
  standup_flag = false;
  sitdown_flag = false;
 }
 if (state == 'D')
 {
  sitdown_flag = true;
  forward_flag = false;
  reverse_flag = false;
  left_flag = false;
  right_flag = false;
  standup_flag = false;
 }
 if (state == 'U')
 {
  standup_flag = true;
  forward_flag = false;
  reverse_flag = false;
  left_flag = false;
  right_flag = false;
  sitdown_flag = false;
 }
 if (state == 'S')
 {
  forward_flag = false;
  reverse_flag = false;
  left_flag = false;
  right_flag = false;
  standup_flag = false;
  sitdown_flag = false;
 }
```

```
  }

  if (forward_flag == true) {
    Forward();
  }
  if (reverse_flag == true) {
    Reverse();
  }
  if (left_flag == true) {
    Left();
  }
  if (right_flag == true) {
    Right();
  }
  if (sitdown_flag == true) {
    sit_down();
  }
  if (standup_flag == true) {
    stand_up();
  }


}
void stand_up() {
  ////////// STAND UP START ////////////////
  Serial.println("up");
  Servo5.write(130);//leg DOWN - Front Left Lift Servo into
  delay(100);
  Servo6.write(130);// leg DOWN - Back Right Lift Servo
  delay(100);
  Servo7.write(130); // leg DOWN - Back Right Lift Servo
  delay(100);
  Servo8.write(130); // leg DOWN - Front Right Lift Servo
  delay(6000);
}

void sit_down() {
  /////// SITDOWN START ////////////////
  Serial.println("dwn");
  Servo5.write(50);//leg UP - Front Left Lift Servo into
  delay(100);
  Servo6.write(50); // leg UP - Back Right Lift Servo
  delay(100);
  Servo7.write(50); // leg UP - Back Right Lift Servo
  delay(100);
  Servo8.write(50); // leg UP - Front Right Lift Servo
  delay(6000);
}
```

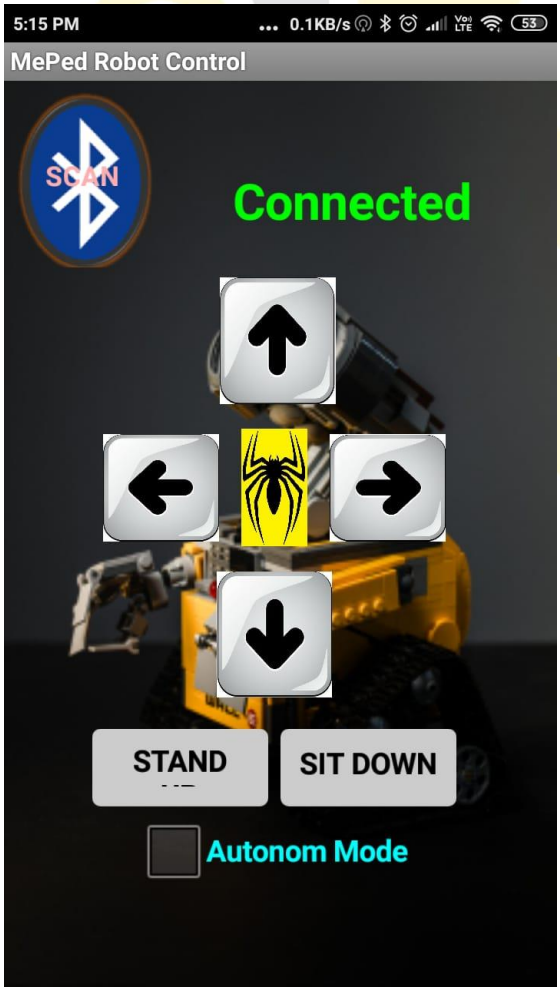
```
void Forward() {
  ///////// RUN FORWARD START ////////////////////////////////
  ///// LEFT SIDE LEGS
  Serial.println("fwd");
  Servo6.write(50); // leg UP - Back Right Lift Servo
  delay(100);
  Servo2.write(30); //right direction - Back Left Pivot Servo
  delay(100);
  Servo1.write(130); //left direction - Front Left Pivot Servo
  delay(100);
  Servo6.write(100); // leg Down -Back Right Lift Servo
  delay(100);
  Servo5.write(50); //leg UP - Front Left Lift Servo
  delay(100);
  Servo1.write(70); // right direction - Front Left Pivot Servo
  delay(100);
  Servo2.write(90); // left direction - -Back Left Pivot Servo
  delay(100);
  Servo5.write(100); // leg Down -Front Left Lift Servo
  delay(100);

  ///////RIGHT SIDE LEGS
  Servo7.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo3.write(150); // left direction - Back Right Pivot Servo
  delay(100);
  Servo4.write(30);// right direction -Front Right Pivot Servo
  delay(100);
  Servo7.write(100);//leg Down - Back Right Lift Servo
  delay(100);
  Servo8.write(50);// leg UP - Front Right Lift Servo
  delay(100);
  Servo4.write(90);// left direction -Front Right Pivot Servo
  delay(100);
  Servo3.write(85); // right direction -Back Right Pivot Servo
  delay(100);
  Servo8.write(100);//leg Down - Front Right Lift Servo
  delay(100);

  ////////// /////// RUN FORWARD END /////////////////
}

void Reverse() {
  ///////// RUN REVERSE START ////////////////////////////////
  Serial.println("rev");
  ///////RIGHT SIDE LEGS
  Servo8.write(50); // leg UP - Front Right Lift Servo
  delay(100);
  Servo4.write(30); //right direction - Front Right Pivot Servo
```

```
  delay(100);
  Servo3.write(130); //left direction - Back Right Pivot Servo
  delay(100);
  Servo8.write(100); // leg Down -Front Right Lift Servo
  delay(100);
  Servo7.write(50); //leg UP - Back Right Lift Servo
  delay(100);
  Servo3.write(70); // right direction - Back Right Pivot Servo
  delay(100);
  Servo4.write(90); // left direction - Front Right Pivot Servo
  delay(100);
  Servo7.write(100); // leg Down - Back Right Lift Servo
  delay(100);

  ///// LEFT SIDE LEGS
  Servo5.write(50);// leg UP - Front Left Lift Servo
  delay(100);
  Servo1.write(150); // left direction - Front Left Pivot Servo
  delay(100);
  Servo2.write(30);// right direction -Back Left Pivot Servo
  delay(100);
  Servo5.write(100);//leg Down - Front Left Lift Servo
  delay(100);
  Servo6.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo2.write(90);// left direction -Back Left Pivot Servo
  delay(100);
  Servo1.write(85); // right direction -Front Left Pivot Servo
  delay(100);
  Servo6.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  ////////// /////// RUN REVERSE END /////////////////
}

void Left() {
  //////// RUN LEFT START ////////////////////////////////
  Serial.println("left");
  Servo5.write(50);// leg UP - Front Left Lift Servo
  delay(100);
  Servo1.write(160); // left direction - Front Left Pivot Servo
  delay(100);
  Servo2.write(90); // middle direction - Back Left Pivot Servo
  delay(100);
  Servo5.write(100);//leg Down - Front Left Lift Servo
  delay(100);

  Servo8.write(50);// leg UP - Front Right Lift Servo
  delay(100);
```

```
  Servo4.write(160); // left direction - Front Right Pivot Servo
  delay(100);
  Servo1.write(90); // middle direction - Front Left Pivot Servo
  delay(100);
  Servo8.write(100);//leg Down - Front Right Lift Servo
  delay(100);

  Servo7.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo3.write(160); // left direction - Back Right Pivot Servo
  delay(100);
  Servo4.write(90); // middle direction - Front Right Pivot Servo
  delay(100);
  Servo7.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  Servo6.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo2.write(160); // left direction - Back Left Pivot Servo
  delay(100);
  Servo3.write(90); // middle direction - Back Right Pivot Servo
  delay(100);
  Servo6.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  ////////// /////// RUN LEFT END /////////////////
}
void Right() {
  ///////// RUN RIGHT START ////////////////////////////////
  Serial.println("rght");
  Servo5.write(50);// leg UP - Front Left Lift Servo
  delay(100);
  Servo1.write(20); // left direction - Front Left Pivot Servo
  delay(100);
  Servo2.write(90); // middle direction - Back Left Pivot Servo
  delay(100);
  Servo5.write(100);//leg Down - Front Left Lift Servo
  delay(100);

  Servo8.write(50);// leg UP - Front Right Lift Servo
  delay(100);
  Servo4.write(20); // left direction - Front Right Pivot Servo
  delay(100);
  Servo1.write(90); // middle direction - Front Left Pivot Servo
  delay(100);
  Servo8.write(100);//leg Down - Front Right Lift Servo
  delay(100);

  Servo7.write(50);// leg UP - Back Right Lift Servo
```

```
  delay(100);
  Servo3.write(20); // left direction - Back Right Pivot Servo
  delay(100);
  Servo4.write(90); // middle direction - Front Right Pivot Servo
  delay(100);
  Servo7.write(100);//leg Down - Back Right Lift Servo
  delay(100);

  Servo6.write(50);// leg UP - Back Right Lift Servo
  delay(100);
  Servo2.write(20); // left direction - Back Left Pivot Servo
  delay(100);
  Servo3.write(90); // middle direction - Back Right Pivot Servo
  delay(100);
  Servo6.write(100);//leg Down - Back Right Lift Servo
  delay(100);
  ////////// /////// RUN RIGHT END /////////////////
 }
```

Step 3: Compile the program and upload to Arduino UNO board.



Thank You !

SUNROBOTICS

www.sunrobotics.in