BAREBONE SV-1201

We help you build your first Industrial Grade Robotic System using our AI powered Development Board and its Supported Peripherals. We create our own smart Servos, DC motors, Stepper Motors as well as Linear Motors.

We are made in INDIA. Software, Electronics and Plastics are also designed and manufactured in India. We are constantly improving our technology and support system to produce better ecosystem in robotics.

BAREBONE is a Series of Smart Motor developed by Elementary Systems to be used in any robotic system using our SDK. We provide all sorts of help and support redardless of the system you are using.

1.Hardware Specification:

Weight	74.5 grams
Dimension	22mm x 53mm x 65mm
Max Resolution and writes/second	1° at 10 writes/second
Stall Torque	12Kg/cm (at 12.0V, 1.5A Max.)
Running Degree	0° ~ 180°
Voltage	9V ~ 12V (Recommended Voltage 11.1V)
Feedback	Current Angle, ID, Torque, Error Flag.
Material	Engineering Plastic + Metal Gears
Features	• Ping • change Id • single motor angle write • Torque Enable/Disable • Read current angle • Write to all motors at Once

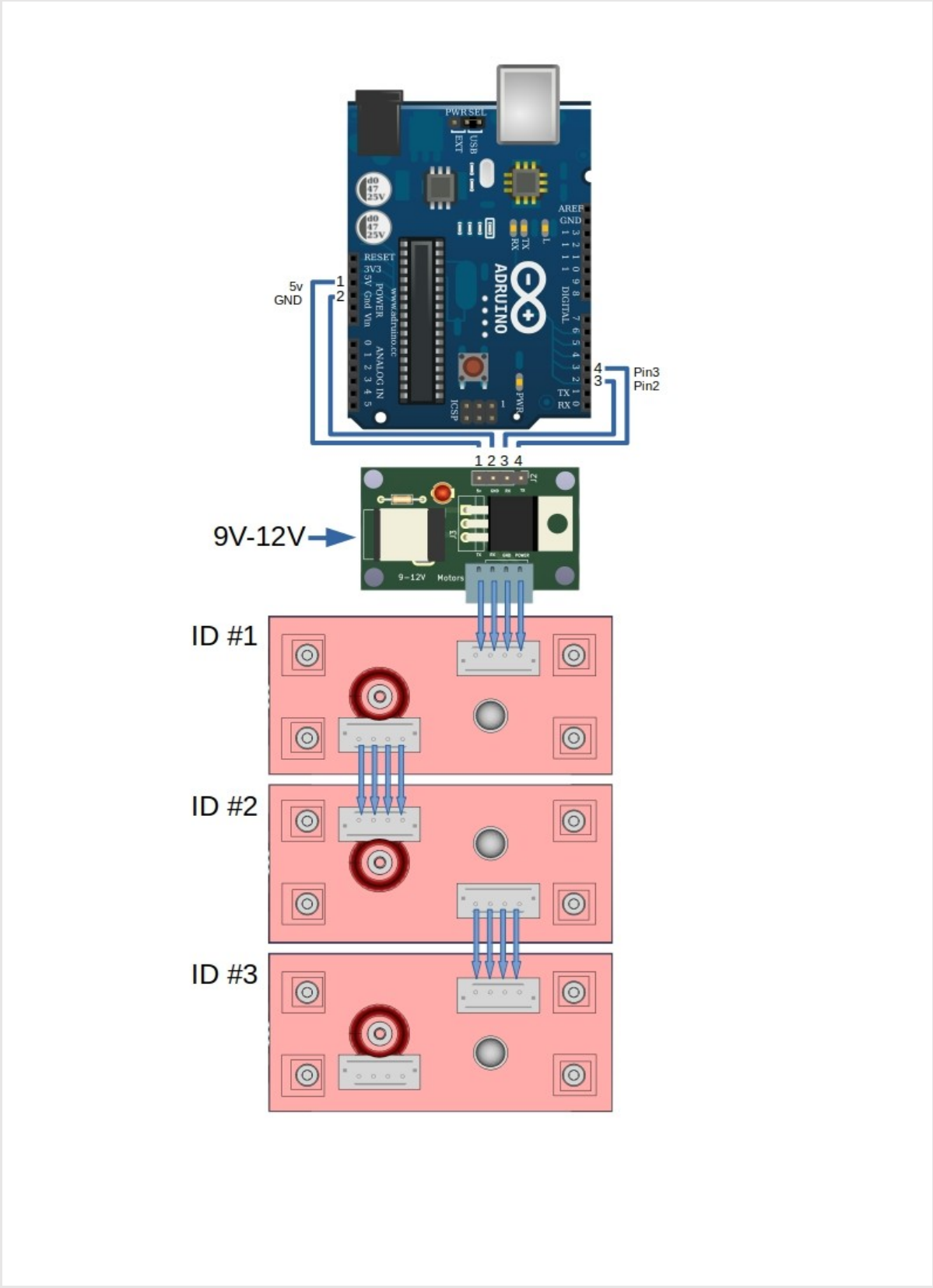
2. Software Specification:

Following commands can be sent from any OS, Evaluation board or controller, given the baud rate is 57600 over serial.

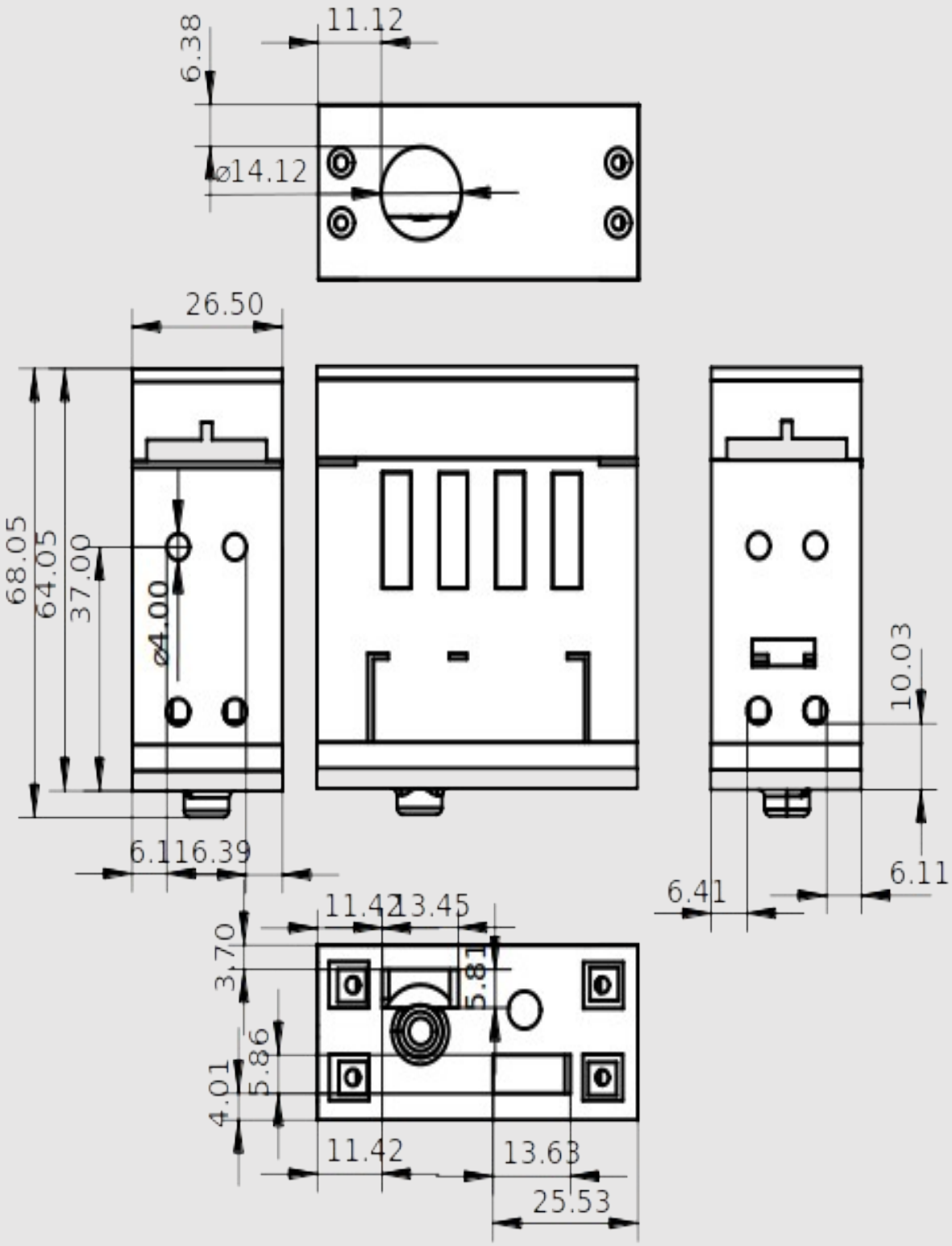
#CMD	Description	Serial Command
Reset	This will reset the ID of the motor to 1. All the motors connected in the rail will get affected by this command. Motor Blinks 5 times on successful write.	255,253,5
Set ID	This command can write a new M**otor ID** to the motors. All the motors connected in the rail will get affected by this command. Developer can have multiple motors of the same ID. Motor Blinks 2 times on successful write.	255,253,1,6,3, MotorID,2, value,0,251
Read Current Angle	This command can help developer retrieve Present Angle of the motors. This command can also be used when the power is disabled. This way developer can record actions sequences from the motors.	255,253,2,4,2, MotorID,16,251
Check Status	This command checks if the given Motor id is connected in the rail. Motor responds back with its id. Motor also Blinks once to indicate itself in the rail.	255,253,1,2, Motorid,251
Write Angle	This command writes the required angle to a motor. Angle can be between 0-180 and in the steps of 5 degrees. This is to individually set the angle of the motor.	255,253,1,6,3, Motorid,13, value,0,251
Write Power	This command asks motors to torque enable or disable. Motor disables the power to the gears when disabled hence makes it easy to move with next to no resistance. User can still read current angle, ping or set id.	255,253,1,6,3, motorid,13, value,0,251
Sync Write	This writes angles to all the connected motors synchronously. Developer just needs to provide the ID's, Angles & total number of motors connected in the rail. Please check the example in code section	Custom packet from SDK

Daisy- Chain Communication:

- Each BareBone SV-1201 can be controlled by sending data packets through a 4-line communication bus.
- Every BareBone SV-1201 can be connected to each other by extending the same link forward.
- Single 4-wire connector is used between the controller and Motor. (Vcc , GND, RX, TX).



CAD DRAFT OF BAREBONE SV-1201



ARDUINO CODE:

```
// Creating Software based Serial to communicate with Motors. Native Serial is still available to users.
#include <SoftwareSerial.h>
// (2 pin) connects to RX pin of breakout board, (3 pin) connects to TX pin of breakout board.
SoftwareSerial mySerial(2, 3);

void setup(){
       // serial used in Motor communication.
       mySerial.begin(57600);
}

void loop(){
       //Put your Code Here.
}

void Ready_to_use_examples(){
       _reset_motors();     /
       uint8_t angle =_ping(1);
       _write_torque(1,0);
       _write_angle(1,90);
       uint8_t angle =_read_angle(1);

       uint8_t MotorIds[]={1,2,3,4,5,6,7,8,9,10};
       uint8_t MotorAngles[]={0,90,0,0,90,90,0,45,135,180};
       uint8_t No_of_Motors=10;


       _syncwrite(MotorIds,MotorAngles,No_of_Motors);
       _set_id(1,2);
}
// This command returns the current angle of the motor is #1.
uint8_t read_angle(uint8_t MotorID){
         uint8_t _return_packet[7];
         uint8_t arrays[]={255,253,2,4,2,MotorID,16,251};
         mySerial.write(arrays,sizeof(arrays));
         delay(100);
         if(mySerial.available() > 0){
                 mySerial.readBytes(_return_packet,7);
         }
         return _return_packet[4];
}
// This command resets the motor id to #1.
void _reset_motors(){
         uint8_t _return_packet[7];
         uint8_t arrays[]={255,253,5};
         mySerial.write(arrays,sizeof(arrays));
         delay(100);
}
// This example is for checking motor id 1 connected or not.
uint8_t _ping(uint8_t motorid){
         uint8_t _return_packet[7];
         uint8_t arrays[]={255,253,1,2,motorid,251};
         mySerial.write(arrays,sizeof(arrays));
         delay(100);
         if(mySerial.available() >0){
             mySerial.readBytes(_return_packet,7);
         }

         if(_return_packet[0] == 255 && _return_packet[1] == 253 &&
         _return_packet[3] == 0 && _return_packet[4] == 0 &&
         _return_packet[5] == 0 && _return_packet[6] == 251){
             return _return_packet[2];
         }
         else{
             return 0;
```

```
        }
}
// This command sets id of the motor from 1 to 2.
void _set_id(uint8_t motorid,uint8_t value){
        uint8_t arrays[]={255,253,1,6,3,motorid,2,value,0,251};
        mySerial.write(arrays,sizeof(arrays));
        delay(100);
}
// This example will command motor #1 to move to 90. (Min-0 Max-180 with of resolution 1 degrees).
void _write_angle(uint8_t motorid,uint8_t value){
        uint8_t arrays[]={255,253,1,6,3,motorid,13,value,0,251};
        mySerial.write(arrays,sizeof(arrays));
        delay(100);
}

// This example will disable power on motor #1. To enable power, _write_torque(1,1).
void _write_torque(uint8_t motorid,uint8_t value){

        uint8_t arrays[]={255,253,1,6,3,motorid,11,value,0,251};
        mySerial.write(arrays,sizeof(arrays));
        delay(100);
}
// This example is for set angle for servo motor ids 1 to 10 in sync mode
void _syncwrite(uint8_t motorids[],uint8_t motor_angless[],uint8_t idsize){
      uint8_t arrays[4+1+(idsize*4)+1];
      uint8_t p;
      uint8_t L_motor_angless=0;
      uint8_t R_motor_angless=0;

      arrays[0]=255;
      arrays[1]=253;
      arrays[2]=4;
      arrays[3]=(1+4*idsize+1);
      arrays[4]=4;
      for(p=0;p<idsize;p=p+1){
            arrays[5+(p*4)]=motorids[p];
            arrays[5+(p*4)+1]=1;

            if(motor_angless[p] > 127){
                  L_motor_angless=127;
                  R_motor_angless=motor_angless[p]-127;
            }
            else{
                  L_motor_angless=motor_angless[p];
                  R_motor_angless=0;
            }
            arrays[5+(p*4)+2]=L_motor_angless;
            arrays[5+(p*4)+3]=R_motor_angless;
         }
      arrays[sizeof(arrays)-1]=251;
      mySerial.write(arrays,sizeof(arrays));
      delay(100);
}
```

Information

This program is created and distributed at **Elementary Systems** and is used to Drive Barebone SV-1201 with all of its functionalities.

Found Problems or got stuck somewhere?


Email Us @ **vaishvik.satyam@elementarysystems.in**
(Reply in 8 Hours, guaranteed)


Contact: **+91-8200186144**
(10AM-6PM, Monday-Saturday)

Documentation and code available at: https://github.com/Elementary-Systems/SV-1201
Video Tutorials of all the features and complications available at: