SVR INFO TECH

Robotics & Automation

RMCS MOTOR KIT

USER MANUAL

www.svrinfotech.net

Table of Contents:

Introduction:

In Education Module, SVR InfoTech offers RMCS motor Kit. This application is simple just by using RHINO DC SERVO DRIVER with an Arduino kit to control RMCS motor respect to the pulse given by the microcontroller which in this case is the Arduino kit. The RMCS motor is connected with the Arduino with an rx tx pins (the RMCS motor use MODBUS UART ASCII communication), while the motor is connected with Arduino which controls the motor with respect to the pulse using a function map in the Arduino IDE.

General Precautions:

Caution: To avoid injury, damage to the robot or equipment, please follow the provided guidelines.

1. Keep away from pets and animals of any kind, animals may behave erratically in the presence of the robot.
2. If the robot is operating abnormally, there is an unusual sound, smell or smoke is detected:
    a. Turn off the robot immediately
3. Always follow the installation and service instructions closely. Keep manuals for future reference.
4. This guide does not cover all possible safety issues or conditions. Always use common sense and good judgment.

5. Please take care of this unit and its accessories, keep them clean. Please do not let this unit or accessories exposed to fire/burning cigarettes, etc... Try to keep the robot and its accessories dry; please do not let this unit exposed to water or moisture.
6. Please do not break, throw or trample the robot.
7. Avoid installation in extremely hot, rainy or water splashing, or being placed in high temperature or moist environment.
8. Please use the accessories provided with this robot.

Components:

MECHANICAL COMPONENTS		
SR. NO.	PART NAME	QUANTITY
1	Plastic Enclosure box (IP65)	1
2	Upper Acrylic Plate	1
3	Bottom Acrylic Plate	1
4	M3*45 (Spacers)	4
5	M3*6 (Nut)	8

RMCS MOTOR KIT

ELECTRICAL COMPONENTS		
SR. NO.	PART NAME	QUANTITY
1	Arduino Uno	1
2	RHINO DC SERVO DRIVER 10V-30V 50W 5A COMPATIBLE WITH MODBUS UART ASCII FOR ENCODER DC SERVO MOTOR	1
3	HIGH TORQUE HIGH PRECISION ENCODER DC GEARED MOTOR 12V 200RPM (RMCS-5014	1

PREREQUISITES		
SR. NO.	PART NAME	QUANTITY
1	Personal Computer with Arduino IDE	1

Connections:

Arduino and **RMCS** connections		
SR. NO.	Touch Sensor pins	Arduino Pins
1	GND	GND
2	TX	RX
3	RX	TX

Code Explanation:

Arduino declaration.

```
byte slave_id=7;                    //Choose the slave id of connected drive.
int INP_CONTROL_MODE=513;           //IMPORTANT: refer datasheet and set value(integer)
according to application
int PP_gain=32;
int PI_gain=16;
int VF_gain=32;
int LPR=334;
int acceleration=5000;
int speed=400;
LiquidCrystal_I2C lcd(0x27,16,2);
long int Current_position;
long int Current_Speed;
long int pules;
 int A=0;
```

Void setup for rmcs motor

```
void setup()
{
   rmcs.Serial_selection(1);       //Serial port selection:0-Hardware serial,1-Software serial
   rmcs.Serial0(9600);             //set baudrate for usb serial to monitor data on serial monitor
   Serial.println("RMCS-2303 Position control mode demo\r\n\r\n");
   lcd.init();
   lcd.backlight();
   lcd.setCursor(4,0);
   lcd.print("WELCOME");

   //rmcs.begin(&Serial1,9600);    //Uncomment if using hardware serial port for
mega2560:Serial1,Serial2,Serial3 and set baudrate. Comment this line if Software serial port
is in use
   rmcs.begin(&myserial,9600);     //Uncomment if using software serial port. Comment this
line if using hardware serial.

rmcs.WRITE_PARAMETER(slave_id,INP_CONTROL_MODE,PP_gain,PI_gain,VF_gain,LPR,accel
eration,speed);    //Uncomment to write parameters to drive. Comment to ignore.
   rmcs.READ_PARAMETER(slave_id);
    delay(3000);
  lcd.clear();

}
  //lcd.setCursor(2,1);
  pinMode(out, OUTPUT);
 //lcd.clear();
}
```

Code for 0 degree to 360 degree.

```
    for ( A=0; A<=360; A++){
     pules=184.166*A;
     Serial.println( "angle : ");
     Serial.println( A);
     lcd.setCursor(0,0);
     lcd.print("Angle=");
     lcd.setCursor(7,0);
     lcd.print(A);

     Serial.print( "pules : ");

     Serial.println( pules);
     lcd.setCursor(0,1);
     lcd.print("pules=");
     lcd.setCursor(7,1);
     lcd.print(pules);

    }
     //delay(2000);

    if(Current_position==66300)
    {
      Serial.println("Position reached.");
       delay(2000);
      lcd.setCursor(0,0);
       lcd.println("Position reached.");
     // lcd.print("Pules=66300");
      lcd.setCursor(0,1);
     lcd.print("CLOCK  A=360       ");
      delay(2000);
      break;

    }
  }

 // delay(2000)
```

Code:

```
  #include<RMCS2303drive.h>


RMCS2303 rmcs;                    //object for class RMCS2303


SoftwareSerial myserial(2,3);     //Software Serial port For Arduino Uno. Comment
out if using Mega.

#include<LiquidCrystal_I2C.h>

//Parameter Settings "Refer datasheet for details" -

byte slave_id=7;                  //Choose the slave id of connected drive.

int INP_CONTROL_MODE=513;         //IMPORTANT: refer datasheet and set
value(integer) according to application

int PP_gain=32;

int PI_gain=16;

int VF_gain=32;

int LPR=334;

int acceleration=5000;

int speed=400;

LiquidCrystal_I2C lcd(0x27,16,2);

long int Current_position;

long int Current_Speed;

long int pules;

 int A=0;


void setup()

{

   rmcs.Serial_selection(1);       //Serial port selection:0-Hardware serial,1-
Software serial

   rmcs.Serial0(9600);             //set baudrate for usb serial to monitor data
on serial monitor
```

```
   Serial.println("RMCS-2303 Position control mode demo\r\n\r\n");

   lcd.init();

   lcd.backlight();

   lcd.setCursor(4,0);

   lcd.print("WELCOME");


   //rmcs.begin(&Serial1,9600);    //Uncomment if using hardware serial port for mega2560:Serial1,Serial2,Serial3 and set baudrate. Comment this line if Software serial port is in use

   rmcs.begin(&myserial,9600);     //Uncomment if using software serial port. Comment this line if using hardware serial.


rmcs.WRITE_PARAMETER(slave_id,INP_CONTROL_MODE,PP_gain,PI_gain,VF_gain,LPR,acceleration,speed);    //Uncomment to write parameters to drive. Comment to ignore.

   rmcs.READ_PARAMETER(slave_id);

    delay(3000);

  lcd.clear();


}


void loop()

{

   Serial.println("Sending absolute position command to -50000");

   rmcs.Absolute_position(slave_id,66300);   //enter position count with direction (CW:+ve,CCW:-ve)


   while(1)       //Keep reading positions. Exit when reached.

   {

      Current_position=rmcs.Position_Feedback(slave_id); //Read current encoder position

      Current_Speed=rmcs.Speed_Feedback(slave_id); //Read current speed
Serial.print(Current_position);
      Serial.print("Position Feedback :\t");
      Serial.print(Current_position);
      for ( A=0; A<=360; A++){
        pules=184.166*A;
        Serial.println( "angle : ");
        Serial.println( A);
        lcd.setCursor(0,0);
        lcd.print("Angle=");
        lcd.setCursor(7,0);
        lcd.print(A);

        Serial.print( "pules : ");

        Serial.println( pules);
        lcd.setCursor(0,1);
        lcd.print("pules=");
        lcd.setCursor(7,1);
        lcd.print(pules);

      }
       //delay(2000);

      if(Current_position==66300)
      {
         Serial.println("Position reached.");
          delay(2000);
         lcd.setCursor(0,0);
```

```
           lcd.println("Position reached.");

        // lcd.print("Pules=66300");

          lcd.setCursor(0,1);

         lcd.print("CLOCK  A=360       ");

          delay(2000);

          break;


       }

    }


  // delay(2000);

  lcd.clear();


   Serial.println("Sending absolute position command to 50000");

   rmcs.Absolute_position(slave_id,-66300);   //enter position count with
direction (CW:+ve,CCW:-ve)


   while(1)       //Keep reading positions. Exit when reached.

   {

      Current_position=rmcs.Position_Feedback(slave_id); //Read current encoder
position

      Current_Speed=rmcs.Speed_Feedback(slave_id);       //Read current speed


       for (int A=360; A>=0; A--){

        pules=184.166*A;


       // Serial.println( "\tangle : ");

        Serial.println( A);

        lcd.setCursor(0,0);
      lcd.print("Angle=-");

      lcd.setCursor(7,0);

      lcd.print(A);

      Serial.println( "pules : ");


       Serial.println( pules);

       lcd.setCursor(0,1);

   lcd.print("pules=");

    lcd.setCursor(7,1);

   lcd.print(pules);


      }




    if(Current_position==-66300)

    {lcd.setCursor(0,0);

       Serial.println("Position reached " );

       delay(2000);

       lcd.setCursor(0,0);

       lcd.println("Position reached.");

  //   lcd.print("Pules=66300 A=0    ");

       lcd.setCursor(0,1);

       lcd.print("ANTI CLOCK          ");

      delay(2000);

       break;


    }


 }
```

```
   //delay(3000);

  lcd.clear();




   Serial.println("Disabling motor.");

   rmcs.Disable_Position_Mode(slave_id);            //Disable postion control
mode

  //delay(1000);


    Serial.println(pules);

}
```

Warranty Terms and Conditions:

Warranty Period: - 90 Days from the date of delivery.

What is covered: - Any Technical defect, malfunctioning.

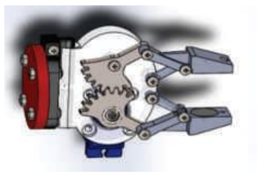
What is not covered: - Physical Damage, Water Damage, Wear & Tear.

What will we do: - Repair or Replacement whichever will be applicable.

Other terms and conditions: -

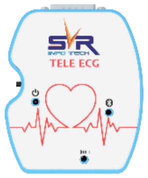
- Warranty will be void if warranty seals are broken.
- Warranty will be void if the product is mishandled.
- Warranty will be void on installation to wrong voltage, overload and wrong application.
- Use other than in accordance to handling instructions.
- Warranty will not extend after replacement.

OUR OTHER PRODUCTS



Hast Articulated Robot Miniature Geared Gripper Miniature Cam

Guided Gripper Miniature Curvilinear Gripper

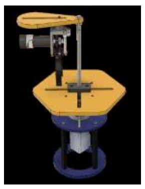


Tele ECG SCARA Robot Flexible Robotic

System Robotic leech

Self-Balancing Robot Ultrasonic Scanner Maze Robot

Digital Dice kit



Multi Gripper Robot Conveyor Belt object counter Line

Follower Robot System Parallel Manipulator Gripper stem

Amber Plaza, 3rd Floor, Near Bank of Maharashtra, Sinhgad College Rd, Ambegaon Budruk, Pune-411046.

Website**: www.svrinfotech.net** E-mail: **admin@svrinfotech.net / support@svrinfotech.net**

+91 9923444362, +91 9975507755 , 020 2984 0013