

Alaonix

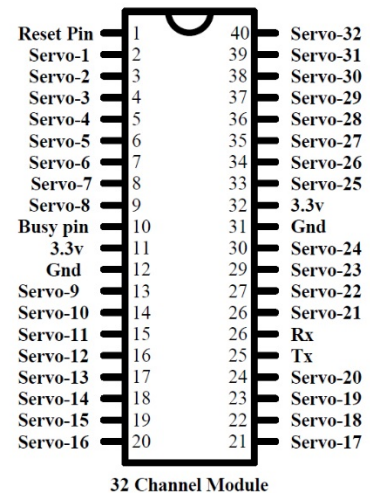
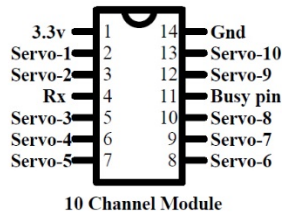
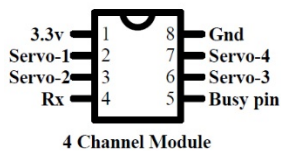
4, 10, 32 Channel Uart Servo Motor Control Module

-- User Manual

Introduction:

There are many robotic projects involve to use servo motor controller module, usually a project needs more servo motors. **Alaonix** built 4, 10 and 32 channel controller modules. They can handle more PWM outputs from 0.5ms to 2.5ms and support more servo motors from 0 – 180°. These modules can help people to build the project himself. Operation command and data can be **Easily** sent to the module via the UART protocol.

Pinout:



Specifications

1. Support 4, 10 or 32 Servo output channels
2. Serial communication interface to MCU or to PC using TTL to USB convertor.
3. 50Hz PWM, 0.5ms – 2.5ms for 0° – 180°.
4. 256 resolution for 0° – 180°.
5. Configurable: Baud rate = 4800, Stop Bit = 1, Data Bits = 8, Parity = none.
6. 3.3V operation (the servos require a 5V supply)

Command and data format:

R = Request command.
Send a char R for request.

S = Servo speed command:

Speed	Level - 1	Level - 2	Level - 3	Level - 4	Level - 5	Level - 6	Level - 7
For Signal Servo	A (65)	B (66)	C (67)	D (68)	E (69)	F (70)	G(71)
For All of Servo	a (97)	b (98)	c (99)	d (100)	e (101)	f (102)	G(103)

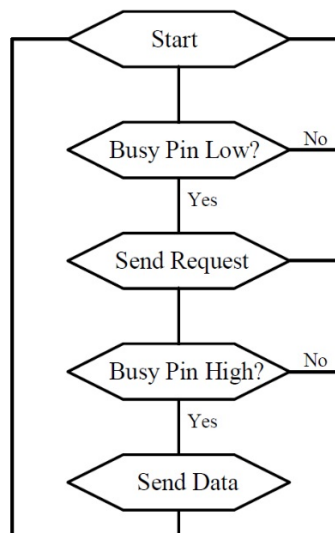
ID = Servo motor ID number.

P = Servo motor position number: 0 – 255 for 0° -180°.

F = Servo speed fine adjust:
 $0 \leq F \leq |D2 - D1|$

E = End of command: # #

Command and Data operation block diagram:



For example:

1: Set all of servo motor to same angle = 90°.

R, (Send knock signal, checking and waiting for busy pin high)
a, n, 127, 1, ##

2: Set ID = 3 servo motor to 180°.

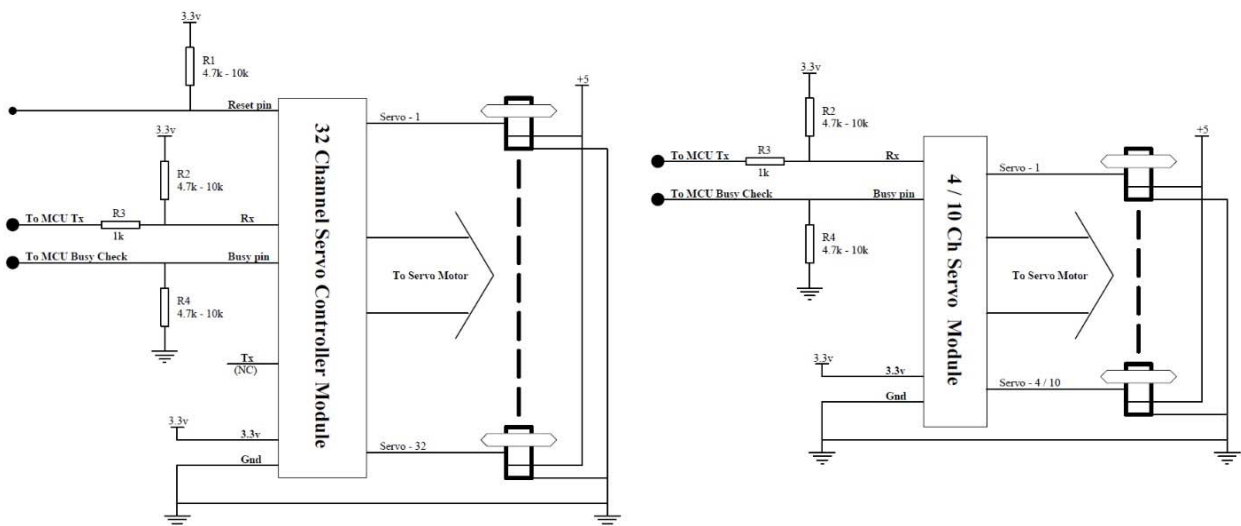
R, (Send knock signal, checking and waiting for busy pin high)
A, 3, 255, 1, ##

3: Set 3 servo motor to different angle: ID = 1, angle = 45°, ID = 2, angle = 180°, ID = 4, angle = 135°.

R, (Send knock signal, checking and waiting for busy pin high)
C, 1, 63, 1, 2, 127, 1, 4, 190, 1, ##

(Please refer our sample code)

4, 10 and 32 Channel Servo Controller module MINIMUM CONNECTION:



Important: The servo controller module PWM output implements timer-based activities, so the servo controller module timer interrupts will be disabled when receiving command and servo speed + position data.

READ THE FOLLOWING TERMS AND CONDITIONS CAREFULLY

LIABILITY WARNING This device should be used only for experimental purposes. It has NOT gone through extensive testing and it could erase or corrupt some or all data on media cards that are inside the device. You assume to take your own risk when you purchase this device, and release the responsibility and liability from the manufacturer with no harm.

REGULATORY WARNING This device is intended solely for experimental purposes; it is not in finished product form and is NOT FCC approved. If you wish to install these modules into nonexperimental final finished products, you will be responsible to have the modules approved by the FCC at your own cost.

Alaonix company

Support: AlaTeam@alaonix.com

Example Arduino Sketch: (Example video: <https://youtu.be/HVmqPbJuco>)

```
/*=====
File.....Alaonix Easy 32 Channel Servo Motor Module
      Test code for Arduino
Purpose.....This test code for Alaonix Easy 32 Channel Servo Motor Module
Board.....Arduino nano
Software....Arduino 1.8.13
MCU.....Arduino ATmega328p
Auther.....Alaonix Team
Website....www.Alaonix.com
Email.....AlaTeam@Alaonix.com
Started....8-25-2020
Updated....
Note: MINIMUM CONNECTION:
      3.3V <=====> Ala Servo Motor Module Vdd
      GND  <=====> Ala Servo Motor Module Vss (Gnd)
      Arduino Tx <=====> Ala Servo Motor Module Rx
      Arduino Pin 2 <====> Ala Servo Motor Module Busy Pin
===== Parameter =====*/

int BusyPin = 2; //Busy check pin
int var = 0;
int ala_id = 0;
int ala_pos = 0;
// For single or all motor
char Ala_Data0[7] = {99, 1, 80, 1, 35, 35, 0};
// For multi Servo motor
char Ala_Data[16] = {69, 17, 30, 1, 18, 30, 1, 19, 30, 1, 20, 30, 1, 35, 35, 0};

void setup()
{
  Serial.begin(4800);
  delay(100);
  pinMode(BusyPin, INPUT); // sets the digital pin 2 as input
  delay(100);
}

void loop()
{
  for (int i = 97; i <= 103; i++) {
    Servo_Speed_Test(i, 30); // zero angle for all servo motor
    ala_delay();
    Servo_Speed_Test(i, 225); // 180 angle for all servo motor
    ala_delay();
  }
}
```

```

Servo_Exercise_Test(66, 30, 17, 18, 19, 20);
Servo_Exercise_Test(67, 30, 21, 22, 23, 24);
Servo_Exercise_Test(68, 30, 25, 26, 27, 28);
Servo_Exercise_Test(69, 30, 29, 30, 31, 32);

Servo_Exercise_Test(66, 225, 17, 21, 25, 29);
Servo_Exercise_Test(67, 225, 18, 22, 26, 30);
Servo_Exercise_Test(68, 225, 19, 23, 27, 31);
Servo_Exercise_Test(69, 225, 20, 24, 28, 32);

Servo_Speed_Test(97, 30);    // zero angle for all servo motor
ala_delay();
Servo_Speed_Test(97, 175);  // zero angle for all servo motor
ala_delay();
Servo_Speed_Test(97, 80);   // zero angle for all servo motor
ala_delay();
Servo_Speed_Test(97, 225);  // zero angle for all servo motor
ala_delay();
Servo_Speed_Test(97, 127);  // zero angle for all servo motor
ala_delay();

Single_Servo_Test(66, 30);
ala_delay();
Single_Servo_Test(71, 225);
ala_delay();
delay(100);
}
//===== Public Delay Function =====
void ala_delay()
{
delay(500);
}
//===== Chack Module busy =====
void Busy_Check()
{
var = 8;
while (var > 0) {
    delay(10);
    var = digitalRead(BusyPin);
    delay(100);}
var = 0;
while (var == 0) {
    delay(150);
    Serial.print("R"); //Send a request and waiting for .....
    delay(100);
    var = digitalRead(BusyPin);
    delay(100);}
}
//===== Servo Speed Test function =====

```

```

void Servo_Speed_Test(int _Speed, int _Angle)
{
    Busy_Check();
    Ala_Data0[0] = _Speed;
    Ala_Data0[2] = _Angle;
    Serial.print(Ala_Data0);
}
//===== Single Servo Test function =====
void Single_Servo_Test(int _Speed, int _Angle)
{
    Ala_Data0[0] = _Speed;
    Ala_Data0[2] = _Angle;
    for (int i = 17; i <= 32; i++) {
        Ala_Data0[1] = i;
        Busy_Check();
        Serial.print(Ala_Data0);}
}
//===== 4 Servo Exercise function =====
void Servo_Exercise_Test(int _Speed, int _Angle, int _ServoID_1, int _ServoID_2, int
_ServoID_3, int _ServoID_4)
{
    Ala_Data[0] = _Speed;
    Ala_Data[1] = _ServoID_1;
    Ala_Data[4] = _ServoID_2;
    Ala_Data[7] = _ServoID_3;
    Ala_Data[10] = _ServoID_4;
    for (int i = 0; i <= 3; i++) {
        Ala_Data[2 + i*3] = _Angle;}
    Busy_Check();
    Serial.print(Ala_Data);
}

```