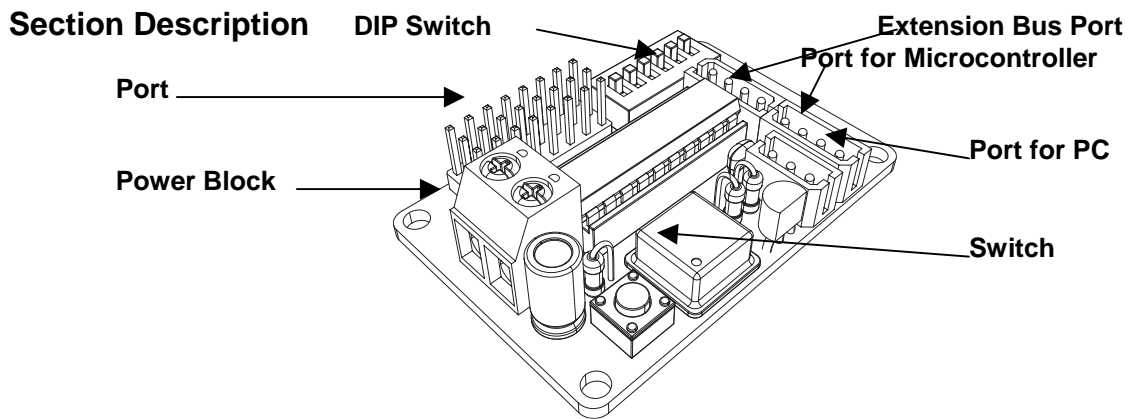
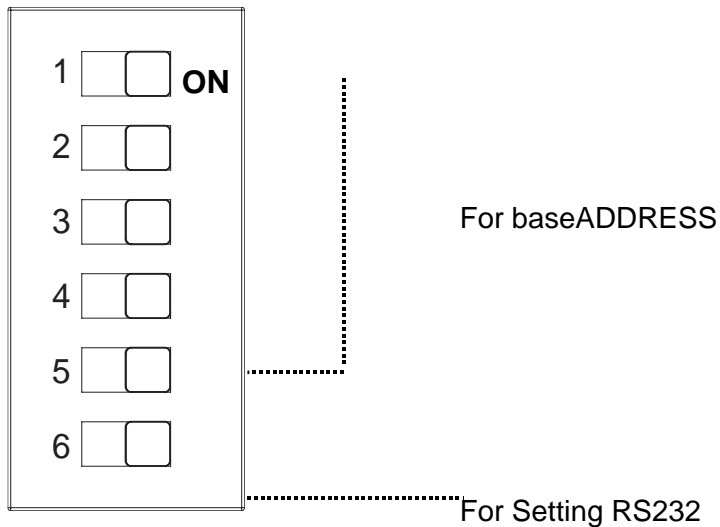


1. Upto 256 RC SERVOs can be connected together using Serial Lines.
2. Angle and speed can be set.
Angle and speed can be set independently set for each SERVO.
3. 4800bps or 9600 bps baud rate are supported (DIP switch setting)
4. SMCpro exclusive cable provided.
(Use with microcontroller, cable for PC must be purchased separately)
5. Free Development tool, SMCpro exclusive Emulator for Windows
 - Button event features can be used.
 - Keyword can be used to make programs.
 - SMCpro can be controlled visually.
 - RC SERVO robot control is easily realized.(Download at our web site)



1. **Power Block(5V)**
Connect power. Please pay attention to + and -. If you connect incorrectly, the unit may be destroyed.
2. **SERVO Port**
Connect to SERVO. 1 SMCpro can connect up to 8 SERVOs.
3. **DIP Switch**
Set baud rate for RS232 and baseADDRESS for SMCpro.
4. **Extension Bus Port**
When connecting multiple SMCpros, you can extend the control bus line with this port. The other SMCpros do not need separate power supply since 5V and GND pins are provided on this port.
5. **RS232 Port (for Microcontroller)**
You can use microcontrollers like PICBasic to control SMCPro. This port uses 5V so it must not be used for PCs.
6. **RS232 Port (for PC)**
You can use PCs to control SMCPro. TXD, RXD, and GND wires are provided to connect to the 9-Pin Serial port.
7. **RESET Switch**
Reset SMCPro

DIP Switch**1. baseADDRESS**

1 to 5 are 5bits for baseADDRESS 0 through 4.

ON: 0

OFF: 1

$\text{baseADDRESS} = [\text{set baseADDRESS value}] * 8.$

baseADDRESS is the lowest base SERVO number of the SMCpro.

The user can connect up to 32 SMCpros together and simultaneously control up to 256 SERVOS. In case this happens, the baseADDRESS is to differentiate the various SMCpros. The factory default value is 00000 or 0h. In other words, the SERVOS will start with number 0.

The set value is in effect once it's RESET.

2. RS232 Baud Rate

SMCpro has two baud rates available, 4800 and 9600bps serial communications.

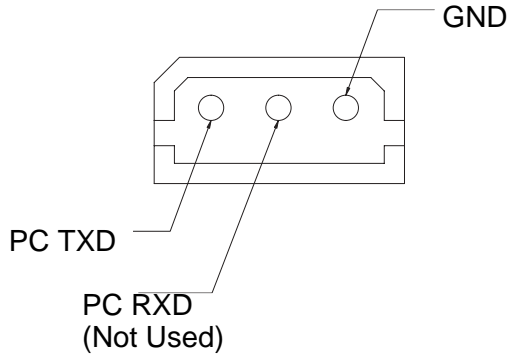
ON: 9600 bps

OFF: 4800 bps

The factory default is 9600.

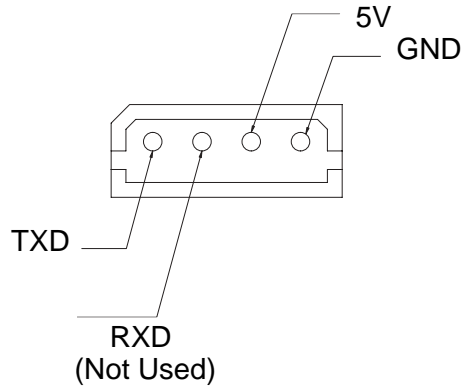
The set value is in effect once it's RESET.

Communication Ports



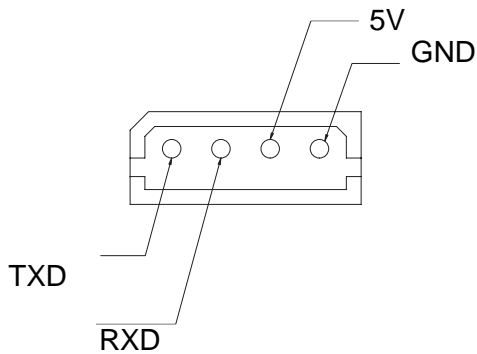
RS232 Port (For PC)

1. RS232 (For PC)
PC's 9PIN Serial Port is used to control the SMCpro, The 3PIN cable is connected here. SMCpro only receives commands from the PC and therefore PC RXD is not used



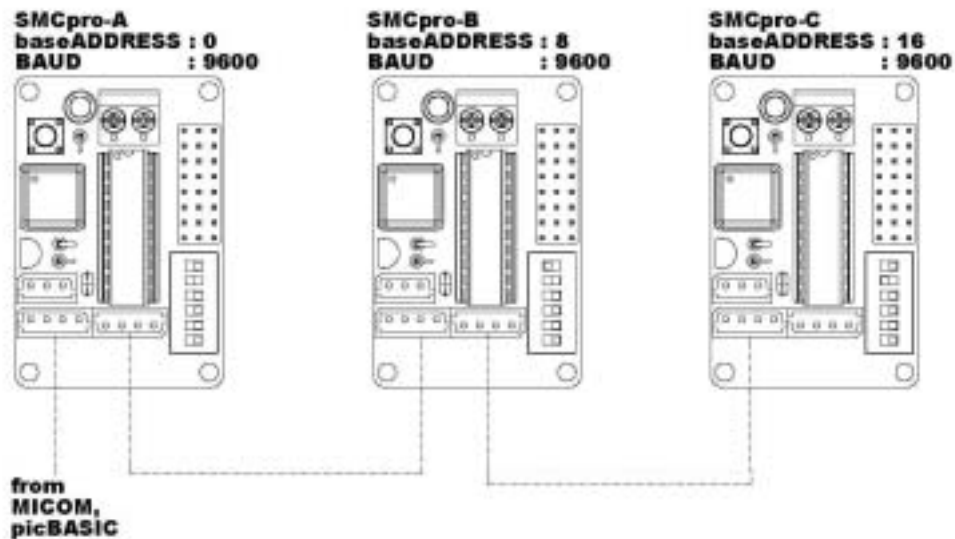
RS232 Port (For Microcontroller)

2. RS232 Port (For Microcontroller)
Microcontrollers such as PICBasic can be used to control SMCpro. This port can also be used as Extended Bus port. 5V is provided in case what you are building is a simple circuit. This port is connected 1:1 with the Extended Bus port



Extended Bus Port

1. Extended Bus Port
Connect multiple SMCpros with this port. This port is connected 1:1 with the RS232(For Microcontroller).



As shown above,
SMCpro-A controls number 0 to 7 SERVOS,
SMCpro-B controls number 8 to 15 SERVOS,
SMCpro-C controls number 16 to 23 SERVOS simultaneously.
In all, one Serial Line can control 24 SERVOS from number 0 to 23.

Please notice how the DIP switches are set in the above diagram. When connecting more than one SMCpro together (multiple SMC pros), please use one BAUD RATE.

Power can just be connected to SMCpro-A and the other units will get power through the extended bus ports.

SMCpro Control

Format: [Servo Number] + [Message] + [DATA]

3 bytes of data are used to control the SMCpro. 1byte takes about 500 microseconds.

1. [Servo Number]

1 byte positive number of the SERVO to send Message

2. [Message]

0 : [DATA] will be value for *speed*.

1 : [DATA] will be value for *angle* (in degrees).

When setting the speed or angle, please be accurate with your numbers since each Servo has independent settings. Through independent settings, a overall smooth operation of SERVOS can be accomplished.

3. [Data]

Either *speed* or *angle* data comes here depending on [Message]

Speed : 0 to 255

Angle : 0 to 180 degrees

```
/* Example in C language*/
...
#define MSG_SPEED 0
#define MSG_DEGREE 1
...
void SerialOut(unsigned char sendVALUE)
{ /* 1 byte positive value is output to RS232 port. */
Output 1 byte positive number;
/* Delay for SMCpro after output.*/
Delay at least 500us;
}
...
void SetSpeed(unsigned char servoNUMBER,unsigned char speedVALUE)
{ /* Set the Speed. All values are positive. (Not characters or strings) */
/* Speed is independently applied. */
/* Only the called Servo is effected by the set speed. */
SerialOut(servoNUMBER); /* Send the SERVO number */
SerialOut(MSG_SPEED); /* Send the speed as DATA */
SerialOut(speedVALUE); /* Send the value of speed between 0 to 255 */
}
void SetDegree(unsigned char servoNUMBER,unsigned char degreeVALUE)
{ /* Set the Angle. All values are positive. (Not characters or strings) */
SerialOut(servoNUMBER); /* Send the SERVO number */
SerialOut(MSG_DEGREE); /* Send the angle as DATA */
SerialOut(degreeVALUE); /* Send the value of angle between 0 to 180 degrees */
}
...
void main(void)
{ ...
...
SetSpeed (3,255); /* Set number 3 SERVO to maximum speed of 255 */
SetDegree(3,45); /* Set the number 3 SERVO to angle of 45 degrees */
/* Once the degree is set, give some time for the SERVO to get to the new position. */
...
...
}
```