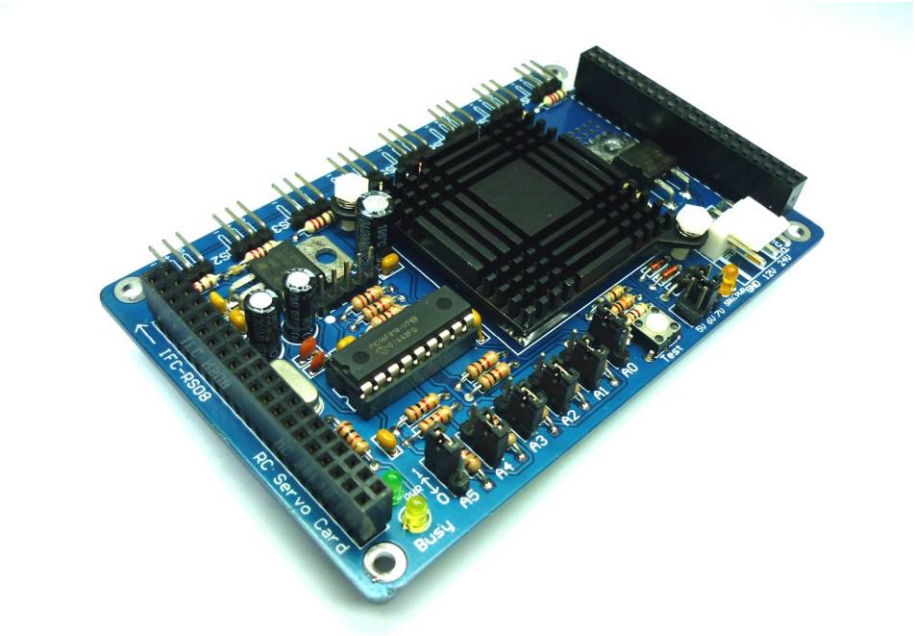
IFC-RS08
Interface Free Controller
RC Servo Card



Card Library Functions

V1.0

March 2009

Function Prototype for RC Servo card (RS08)

This document explains the function prototype for RS08 needed in controlling servo motors. The function prototype will be called in main program for MB00 in order to control/communicate with RS08. User can also find the explanation of function prototype in its header file, "iic_rs.h". Table 1 shows the function prototype for RS08, the example card address used in the function prototype is 'add_rs'.

Function Prototype	Remarks	Parameter Description
void rs_en(unsigned char **add,** unsigned char **servo_ channel,** unsigned char **enable)**	rs_en(**add_rs,5,1**)	**add** = card address **servo_channel** = 0-8 RC servo motor's channel 0 – All RC servo's channel 1 – 1st Channel 2 - 2nd Channel 3 – 3rd Channel 4 – 4th Channel 5 – 5th Channel 6 – 6th Channel 7 – 7th Channel 8 – 8th Channel **enable** = Enable or disable RC servo's channel 1 – Enable 0 – Disable
void rs_pos_sp(unsigned char **add**, unsigned char **servo_channel**, unsigned int **pos**,unsigned char **ramp**)	rs_pos_sp(**add_rs, 5, 4500,20**)	**add** = card address **servo_channel** = 0-8 RC servo motor's channel 0 – All RC servo's channel 1 – 1st Channel 2 - 2nd Channel 3 – 3rd Channel 4 – 4th Channel 5 – 5th Channel 6 – 6th Channel 7 – 7th Channel 8 – 8th Channel **pos** = 0 – 5000 steps RC servo's position 0 – 0.5ms pulse width 5000 – 2.5ms pulse width **ramp** = 0-255 speed rate 0 - move as fast as possible 1 -slowest in speed rate 255-fastest in speed rate

unsigned int rs_read_pos(unsigned char **add**, unsigned char **request_servo_channel**)	rs_read_pos(**add_rs,3**)	**add** = card address **request_servo_channel** = 1-8 RC servo motor's channel 1 – 1st Channel 2 – 2nd Channel 3 – 3rd Channel 4 – 4th Channel 5 – 5th Channel 6 – 6th Channel 7 – 7th Channel 8 – 8th Channel **Return value =** 0-5000 steps requested RC servo's position. 0 – 0.5ms pulse 5000 – 2.5ms pulse

Table 1 Function Prototype for RC Servo Card (RS08)

RC Servo Card (RS08) is another card for IFC. With this **RC Servo** card user can plug their favorite servo motors and use them. Examples in using function prototype are given as a guide.

void rs_en**(**unsigned char **add,** unsigned char **servo_ channel,** unsigned char **enable)**

IFC-RS08 has 8 channels which can drive RC servos independently. And this function is used to enable or disable the servo channel that want to use. When power up the IFC-RS08, RC servos are at the loose condition as they are initially disabled by the RS08. Therefore, user needs to activate or enable the RC servo's channel. Any RC servo will not work even the correct pulse is given from MB00 unless the channel has been enabled. The servo motor initially will turn to 2500 steps (1.5ms pulse width) right after the RC servo is enabled if there was no pulse given from MB00 before. If the pulse has been given, lets say with 1ms pulse width then the servo motor will turn to that position right after enabled. **add** is refer to address used and this address must compatible with address at IFC-RS08 card. **servo_channel** represent RC servo's channel. Given value 0 for servo_channel means enable the entire eight channels. **enable** is for enable or disable the RC servo's channel. Given value 1 is to activate or enable the selected RC servo's channel, while value 0 is disable. User is free to enable or disable either all or any one of the RC servo's channel during the programming.

This is an example to call this function. Regarding on this function, all RC servo's channel will be enable. Thus, all RC servo will turn to a position according to their own signal pulse.

rs_en(**add_rs,0,1**) //enable all RC servo channel

void rs_pos_sp(unsigned char **add**, unsigned char **servo_channel**, unsigned int **pos**,unsigned char **ramp**)

This function is used to determine channel, position and speed for RC servos. In other words, this function will generate signal pulse to RC servos. Servo motor will turn to the given position if that channel has been enabled. **add** is refer to address used and this address must compatible with address at IFC-RS08 card. **servo_channel** represent RC servo's channel. Since IFC-RS08 has 8 channels, user may use **servo_channel** to define which RC servo's channel that user use. If user would like to control the entire channel to the same position then value of 0 should be given. For value more than 8, IFC-RS08 will determine it as 8th channel. **pos** is for RC servo position. Position value for IFC-RS08 is from 0 steps to 5000 steps and the initial position is at 2500 steps. If servo position is more than 5000 steps, IFC-RS08 will determine it as 5000 steps.

Pulse width	0.5ms	1ms	1.5ms	2ms	2.5ms
Steps	0	1250	2500	3750	5000

Table above shows the pulse width for selected steps. For example, a position of 0 steps actually demands the RS08 to give a continuous pulse width of 0.5ms at 20ms period to the selected RC servo's channel. From the explanation in IFC-RS08 User's Manual, the servo motors initially is at the position of 2500 steps as the signal pulse given to each RC servo's channel is 1.5ms. If user requests position more than 2500 steps, RC servo will turns to one side and if user request position less than 2500 steps, RC servo will turns to another side. Before that, the servo motor's channel must be enabled first. **The exact correspondence between pulse width and position is varies from one servo manufacturer to another**. **1.5ms is not necessarily neutral or middle position. It is depends on what types or brands RC servo user use.**

ramp represent the speed rate for RC servo go to a position. The speed rate is from 1 – 255. User may choose one of 255 ramp rate (speed rate) for each servo. The higher value, the faster servo will rotate to its position. However, value of '0' will disable the speed, thus provide normal speed, the servo motor will rotate according to it own maximum speed. Decimal value '1' indicates that the servo will run at slowest speed and decimal value '255' will run at fastest speed. At each 20ms interval, the current servo position will increase or decrease with the speed value depending on whether the position is greater or lesser than the new position. The speed of servo motor is depending on the brand of servo motors. Therefore, different type of servo would result in different speed and position.

This is an example to call this function. The initial position for RC servo is 2500 steps. **add_rs** is refer to address used and this address must compatible with address at IFC-RS08 card. When users call this function, RC servo will turns to position of 4500 steps with a ramp rate of 20. The position either clockwise or counter clockwise is depends on the RC servo used.

rs_pos_sp(**add_rs,5,4500,0**) //drive RC servo channel 5 to 4500 steps as fast as possible

unsigned int rs_read_pos(unsigned char **add**, unsigned char r**equest_servo_channel**)

This function is to read the position of requested RC servo channel. This function is usually use when the user wants to know what is the current position of one RC servo. **add** is refer to address used and this address must compatible with address at IFC-RS08 card. **Request_servo_channel** represent RC servo's channel. If the value of 0 is given, IFC-RS08 will recognize it as 1st channel. For value more than 8, IFC-RS08 will determine it as 8th channel. Return value is the current position for selected RC servo. The value is from 0 step to 5000 steps.

When users call this function in program, IFC-MB00 will read the requested RC servo's position from RC servo channel. **add_rs** is refer to address used and this address must compatible with address at IFC-RS08 card. From example below, IFC- MB00 read the RC servo's position for channel 8 from IFC-RS08. The return value for this function is the requested servo's position which is 0-5000.

```
Pos = rs_read_pos(add_rs,5)        //read position of RC servo's channel 5
                                   //the value 0-5000 is stored in variable pos.
```

Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: +607-521 3178
Fax: +607-521 1861

URL: www.cytron.com.my
Email: support@cytron.com.my
sales@cytron.com.my