



RC (Radio Controlled) servo motors are small and relatively cheap motors with built in gearing and feedback control loop circuitry. They are mainly used in model building (cars, boats, etc.), but they also would be very useful in combination with the Robo Interface. It is however not possible to connect RC-servo motors directly to the RoboIF and to control them with the RoboPro software.

Here a simple interface is presented that can perform that task. Up to 8 servos, each with 7 positions, can be connected to the RoboIF and controlled with RoboPro. Servos with 9 positions are also possible. The positions range from 30 % to 90 % (or 10 % to 90 % for the 9 position servo) in steps of 10 % of the total range of the servo.

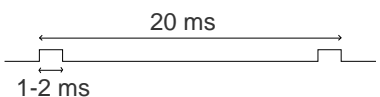
The circuit is made using only 5 simple (and cheap) chips from the CD4000 logic series.

Knobloch sells servos with some helpful things (part# 132292K, 132290K and 132004K) very well suited for Robo models. Conrad also has many servos. I used some of them (the cheap Topline servos) for testing the interface and they worked well.

### RC servo motors

These motors have the following properties:

- Most of them rotate from 0° to 180°
- All have a three wire connection:
  - black ground (-)
  - red power (+)
  - white/yellow signal



The connectors are however different for different brands

- The power supply must be 4.8 V to 6 V

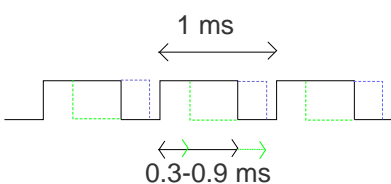
The control signal with which the servo arm is positioned must be a pulse of 1 ms to 2 ms every 20 ms. 1 ms means that the servo arm is in the left position (0°) and 2 ms that it is in the right position (180°). And of course any position in between can be made: e.g. 1.5 ms = 90°.

### The outputs of the RoboIF

The outputs O1 to O8 make use of PWM (pulse width modulation) for the intensities 1 to 7. Every ms they switch on and off for some time, 0.3 ms on for intensity 1 and then in steps of 0.1 ms to 0.9 ms for intensity 7.

intensity on off (in ms)

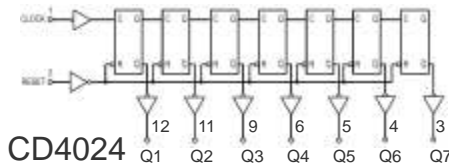
1	0.3	0.7
2	0.4	0.6
3	0.5	0.5
4	0.6	0.4
5	0.7	0.3
6	0.8	0.2
7	0.9	0.1



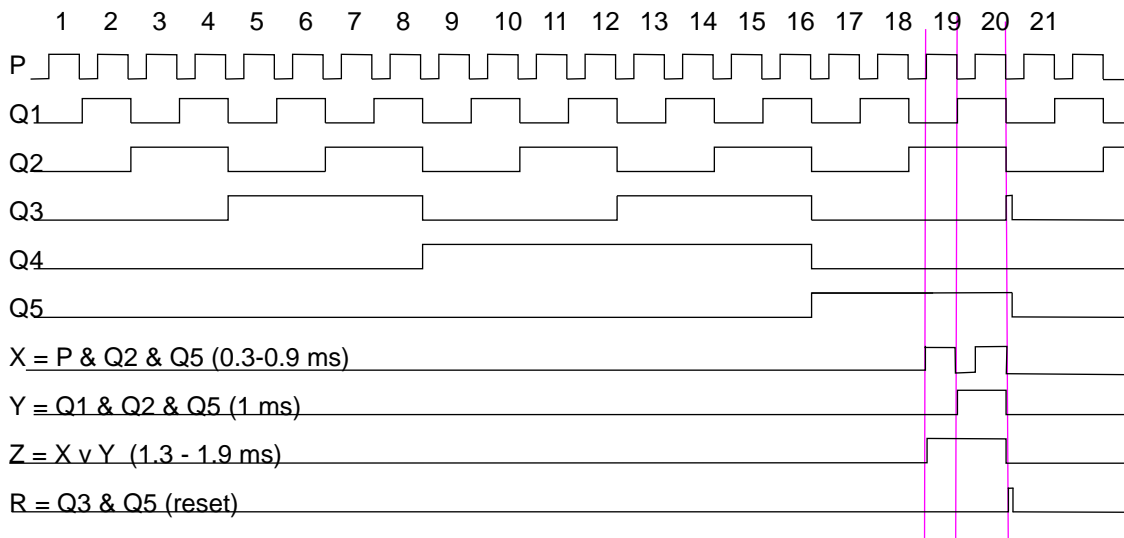
It is the task of RoboIF to RCservo interface to make from these pulses the signal that is needed for the servo.



### Basics of the circuit



The basic idea for the RoboIF to RCservo interface makes use of the properties of the outputs of the RoboIF. If we can "glue" together one period (= 1 ms) of the PWM signal with one pulse (= 0.3 ms to 0.9 ms), we get a pulse of 1.3 ms to 1.9 ms, and that is just what is needed for the servo motor. This pulse is only needed once in 20 ms, so we count 20 pulses and then start again. For the counter the first 5 stages of a CD4024 7-stage ripple counter are used. This counter counts at the negative going transition of the input pulse.



The timing diagram shows how this is all accomplished. P is the input pulse of one of the RoboIF outputs. After the 18th pulse Q2 and Q5 are both high and there we catch input pulse 19 (and 20, X) and Q1 (Y). Z is the combination of these two signals and is what is needed to control the servo.

- (1)  $Z = X \vee Y$
- (2)  $Z = (P \& Q2 \& Q5) \vee (Q1 \& Q2 \& Q5)$
- (3)  $Z = (P \vee Q) \& (Q2 \& Q5)$
- (4)  $Z = \overline{(\overline{P \vee Q}) \vee (\overline{Q2 \& Q5})}$

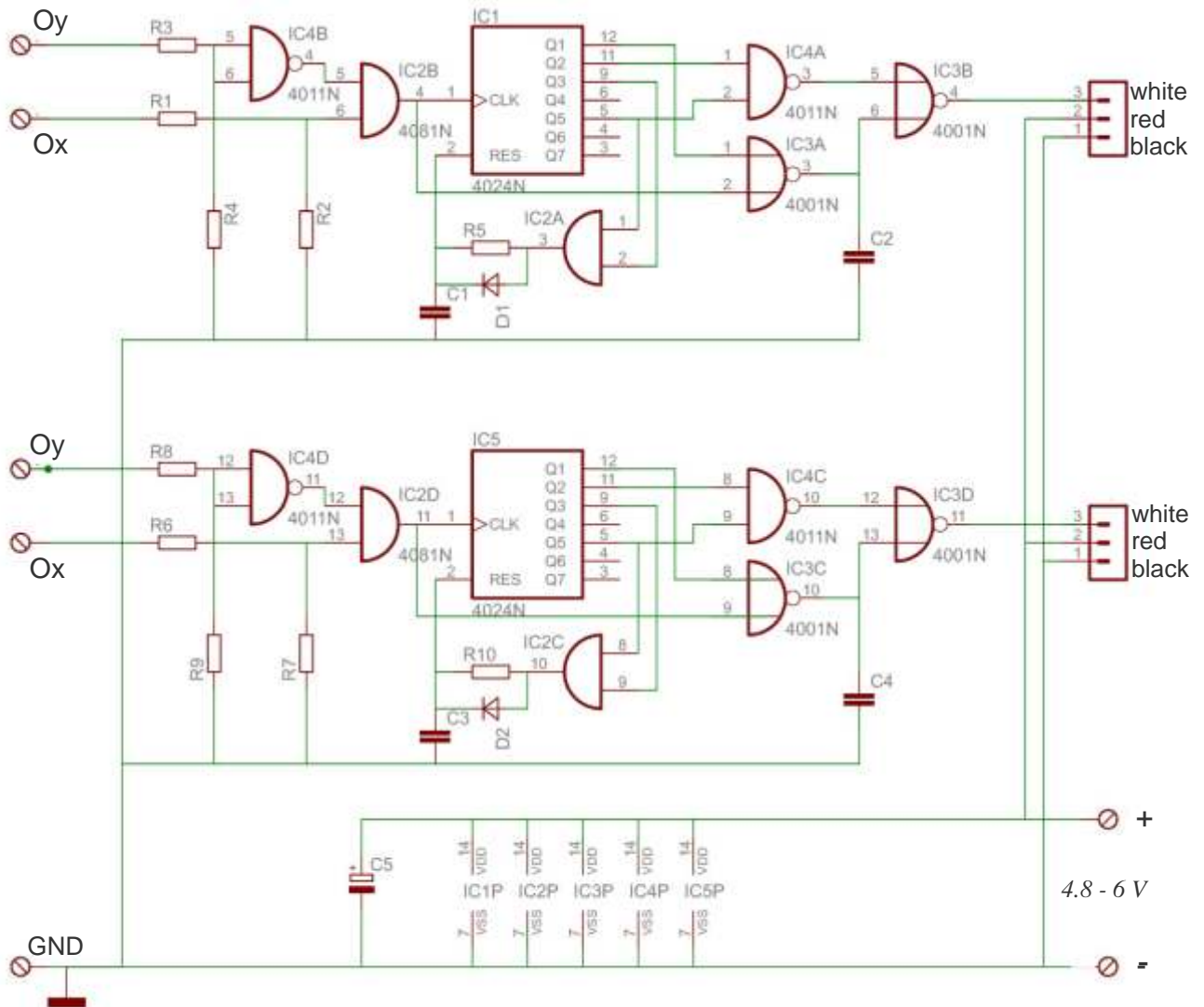
- (1) After the 20th pulse the counter must be reseted, so the cycle can start again. This is done by Q3 & Q5 (R) which goes high for the first time after 20 input pulses.
- (3) The logic (2) can be simplified with boolean algebra to (3).
- (4) Since we need an inverter for the 9 position logic it appeared to be convenient to change (3) to (4) with Morgan's theorem.

### 9 positions

The 7 possible positions can be extended to 9 positions if a second output is used. Look at the table on the previous page. If we invert the signal we get with the intensities 6 and 7 pulses of 0.2 ms and 0.1 ms and these form a nice addition to the pulses of 0.3 ms to 0.9 ms we already have. But two outputs instead of one are needed.

### Two servos

With just one CD4024 counter more it is possible to make the circuit suited for two servos. There are just enough gates left.



ip01 - RCservo to RoboF interface 1.0

### The circuit diagram

IC1 , IC5 : CD4024 (7-stage ripple counter)  
 IC2 : CD4081 (quad 2-input AND)  
 IC3 : CD4001 (quad 2-input NOR)  
 IC4 : CD4011 (quad 2-input NAND)

R1, R3, R11, R13 : 10k  
 R2, R4, R12, R14 : 100k  
 R5, R15 : 47 k  
 C1, C2, C3, C4: 47 pF  
 D1, D2 : 1N4148  
 C5 : 10  $\mu$ F

1. The circuit may get its power from a net supply or from accus or batteries. No provision has been made for wrong poling, because a diode would cause loss in voltage. So be careful when you connect the power.

2. The resistances at the input (R1/R2 and so on) have three functions: they divide the 9 V of the interface by 2, unused inputs are tied to ground and they protect the inputs of the chips against high currents.

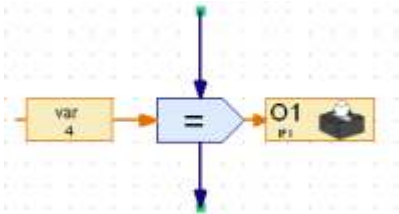
3. The R5/C1/D1 (and R10/C3/D2) circuit is needed for making the reset pulse somewhat longer and for a reliable reset of all the counter stages.

4. There is a very small delay between the down going P signal and the up going Q1 signal. C2 and C4 bridge this little gap.

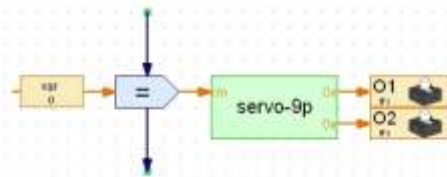


**Software**

Using and programming the servos is very simple in RoboPro.

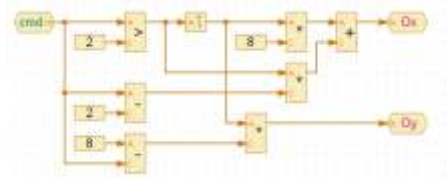


For the 7 position servo only the Ox input is used, Oy must be left unconnected. Then just send a command value of 1 to 7 to the output.



For the 9 position servo both the Ox and Oy inputs are needed.

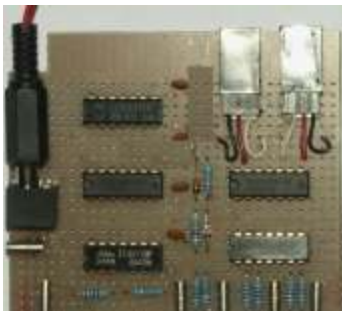
For the first two positions Ox (O1 in the example) must be set full on (intensity 8). Oy (O2) can have the values 6 and 7.



For the other 7 positions Oy (O2) should be set off (intensity 0). Ox (O1) can have the values 1 to 7.

This simple routine takes care of these requirements and numbers the positions neatly from 1 to 9.

Have a look at the two test programs to see how it works.



Here the circuit has been built on a piece of prototype circuit board. It looks somewhat clumsy (due to some changes afterwards), but it works very well!

**More information**

About servo motors:

<http://www.societyofrobots.com>

<http://www.roboternetz.de> (in german)

RC-servo motors;

<http://www.knobloch-gmbh.de>

<http://www.conrad.de>

Datsheets of the CD4000 series

<http://www.alldatasheet.com>

Programming in other languages:

<http://www.ftcomputing.de>