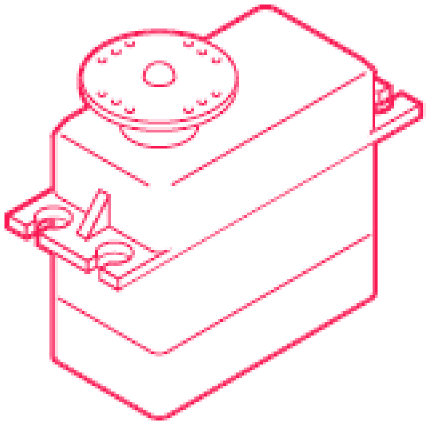
Servos with a Smile



Nick Baldwin worked with Nottingham Trent University Engineering Teacher trainees this year developing servo engineering using PICs and timer chips. This feature provides a useful basis for developing your own engineering approach to using servos in projects, assignments and workshop based experiments at level 2 and 3.

Introduction

Students and teachers in Engineering and Design and Technology rarely explore electro mechanical work outside using DC permanent magnet motors. These have poor stall / start characteristics, lack elegance and still require a gearbox! Where projects generally move away from permanent magnet motors it is usually with large, relatively heavy and slow stepper motors.

Accurate movement of devices driven by electric motors requires sophisticated feedback circuits or the use of such stepper motors. However an entirely different class of motor the servomotor or servo is an ideal prime mover for a wide range of motorised applications and functions. Long used in radio control models and as remote actuators in the film industry, servos provide good torque speed position characteristics and remain fixed at a number of angular positions by a particular signal pulse sent to them.

Background to R/C Servos

Servos have been used by modellers for many years to accurately control their models whether they are land, water or air based.

This example shows two servos being used to control the rotors on an electric helicopter.

Servos have not been used by large numbers of students in schools probably due to their cost. Like everything else servos have improved over the years and the cost is now easily within reach of most student budgets. With only a three wire connection, lightweight and small size they are now set to be a popular lower cost alternative to stepper motors.

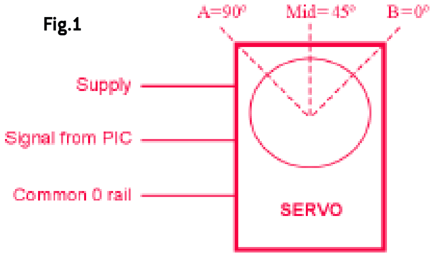
They are ideal in applications in project work without the frustrations of additional gearboxes and control circuits. The project work around them can combine all the construction skill and the demands of working with sheet metal and plastics, moulding or tubing and light fabrication.

Small servos can be bought from model shops for just £4 but the one below was obtained off an internet site for 99p!

Technical

Servos are a means of very accurate position control and have certain advantages over stepper motors. Servos are a very accurate motor and gearbox assembly that can be repeatedly moved to the same position due to their internal position sensor. They are operated by pulse widths of **0.75** to **2.25ms** every **20ms**; and this typically gives between **90°** and **120°** rotation. Giving a pulse of **1.50ms** would send the servo to its mid point. An advantage of servos over stepper motors is the ease with which they can be programmed and the fact that they do not over-run and loose their datum point as stepper motors can, especially if rotated quickly. A downside is the servo can only be programmed in basic but the commands are simple and are not beyond a Key Stage 4 student studying Systems and Control where they have to include a mechanical output.

Fig.1



Servos have 3 wires: Power, Ground and Signal (Control) they Use Pulse Code Modulation (PCM) on the signal wire to control the output shaft position. Servos are a good example of a **closed loop feedback system.** A small potentiometer is internally connected to the output position shaft inside the servo and its resistance is proportional to the servo position. The internal comparator electronics on-board the servo uses this as the error or feedback signal and compares it with the incoming signal generated using a pulse stream from R/C gear, 555/6 timer, a PIC controller or even a computer port.

The pulse stream is in 20ms blocks (that is 50 pulses per second or 50 Hz). The width of the pulse determines the servo position as this pulse is converted in the servo and compared with the internal error signal. Effectively we are saying that the 20ms pulse stream is a carrier wave modulated (changed) by altering its width.
The actual width of the pulse each 20ms is quite small, between 0.75ms and 2.25ms and these are typical for most servos.

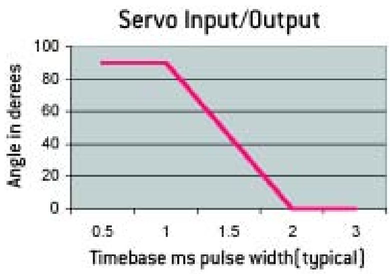
Remember servos require a constant pulse stream not just single pulse.

These width values give the limits of feedback correction signal so:

0.75 ms – 90° Position on servo arm
1.25 ms – 45° Position on servo arm
2.25 ms – 0° Position on servo arm

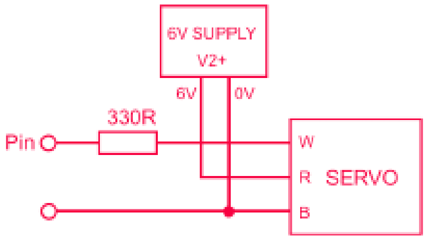
See Fig.1

This is useful as the arm position can be fine controlled to any position between those two limits accurately by any pulse value between those limits. Repeated every 20ms will hold that position until a new value is set or created on the signal wire. If the pulse is lost the servo will lose its position.



Graph showing relationship between servo arm position and pulse width

Servos are quite 'noisy' devices and require relatively large currents so in most cases they require their own power supply with the 0v rail connected (commoned) to the PIC 0v rail.



Note: the servo requires a separate power source as 'noise' and load demand may upset a sensitive circuit like a PIC or change voltage levels on a 555 / 556. Current draw from servos does have an impact on battery life but only if there is a high duty cycle.

Control Methods

We will ignore the R/C method of control as although increasingly of low cost and great for amateur 'roboteers' and model flyers it does little for the engineer. R/C is self contained and ready to run on two servos for less than £30 and is ideal for 'test bedding' a mechanical system. Integrating servos into engineered products demands a more detailed control approach.

The two methods explored here are using a PIC in this case, PICAXE programmed to suit and a dedicated modulated astable or 555 timer for short (now you can use that stock of 555's up on really interesting work!)

Basic PIC Routine

Using the PICAXE system any of the following PICs can use the servo or pulsout command. (08, 18,18x, 28A, 28x, 40) the servo command sets up an internal routine.

The simplest routine would be:

TEST ROUTINE

```
loop: servo 1, 75
      pause 2000
      servo 1, 150
      pause 2000
      servo 1, 225
      goto loop
```

This simple programme would move the servo to end A for 2 seconds, then the mid position for 2 seconds and then the other end B for 2 seconds, the process then repeated in a loop. This would be a good test programme to fine tune a PIC and servo of unknown characteristics as you can 'fine tune' the servo position values and test the positional limits of the servo

The syntax described as servo pin, pulse allocates pin in/out number followed by pulse width. So servo 1, 75 is pin 1 pulse width 0.75ms. The servo command instructs the PIC to generate this pulse every 20ms or at 50Hz. This servo command in BASIC is unlike other commands in that the instruction continues until superceded by another servo command.

Some student work focused not just on line positional programming using basic but included analogue in processing. Entry level PICs have low resolution analogue to digital conversion so are less suitable and of course limit the number of lines of programme.
Note: servo commands can only be programmed in basic, not using a flowchart.

A simple forward reverse (left/right) 2 button control using pin 6 and pin 7 inputs (PICAXE 18) would be:

FORWARD REVERSE

```
main:
label_1B:   if pin6=1 then label_34
            if pin7=1 then label_3B
            servo 1,150
            pause 1000
            goto label_1B
label_34:
            servo 1,75
            pause 1000
            goto label_1B
label_3B:
            servo 1,225
            pause 1000
            goto label_1B
```

Simply described this routine scans every second, pin 6 and 7 looking for a high command input and then directs that instruction to servo and then loops to the start of the routine.

ANALOGUE INPUT PROGRAM

Using an analogue input program using a potentiometer input Note: high values for adc and number of program lines.

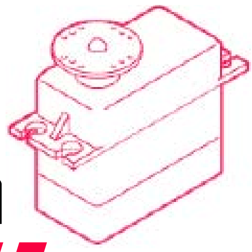
```
main:
label_17:   readadc 0,b0
            if b0> 100 then label_101
            servo 7,75
label_13D:  pause 100
            goto label_17
label_101:  if b0> 120 then label_4E
            servo 7,90
label_15E:  pause 10
            goto label_17
label_4E:   if b0> 130 then label_59
            servo 7,120
            goto label_15E
label_59:   if b0> 140 then label_6F
            servo 7,150
            goto label_15E
label_6F:   if b0> 150 then label_85
            servo 7,175
            goto label_15E
label_85:   if b0> 160 then label_90
            servo 7,190
            goto label_13D
label_90:   if b0> 170 then label_9B
            servo 7,210
            goto label_13D
label_9B:   if b0> 190 then label_A6
            servo 7,225
            goto label_13D
label_A6:   if b0> 200 then label_13D
            pause 300
            goto label_17
```

This programme reviews line by line the ADC values of 0, b0 and update the servo position continuously.

Continued overleaf

Servos with a Smile

Continued



Using 555 timers as PWM inputs

Budget controllers that work really well for analogue and push button position can also be built using a pair of 555 timer chips or a 556 pair.

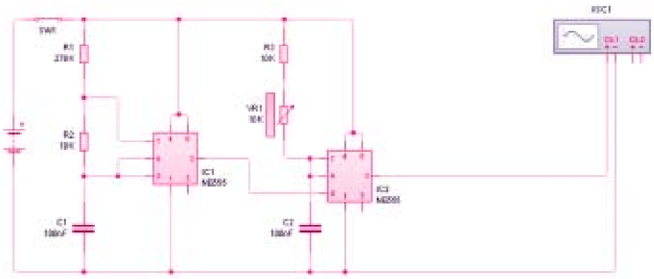
How it works:

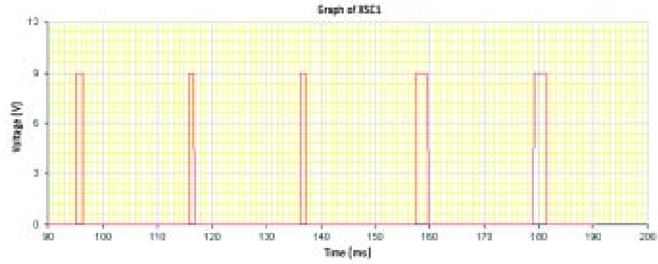
IC1 (555) – is configured (wired) as an astable multivibrator and this produces continuous pulses of around 20ms that is 1/cycles= frequency eg: 50 Hz. This can be fine tuned to precisely 20ms using VR1 and an oscilloscope. In practice we did not need to – as luck would have it, this circuit works out nicely with standard component values (10k, 270k and so on) so production and reliability really is simple.

The resultant output on pin 3 of IC1 is effectively an on/off switch for IC2 (configured as a monostable multivibrator) that is then 'enabled' on it's trigger pin 2 so that the input pin 6 or threshold pin can effectively turn on a pulse within the 20ms time frame. This with a fixed duration in our case between 0.75-2.25ms to drive a servo, the actual width is controlled by VR2 and can be calibrated.

This worked straight off the drawing board for students who in many cases used this as a starting point before moving on to PIC work as they then understood the signal generation and processing going on.
Note: we used 10K variable resistors with linear tracks.

Amazingly the small servos we used worked well without a power transistor and we stuck with that, but good practice suggests the output pin 3 is fed into a power transistor to provide stable switched control. Note the output is now inverted so the servo will operate clockwise /anticlockwise inversely to the input signal.

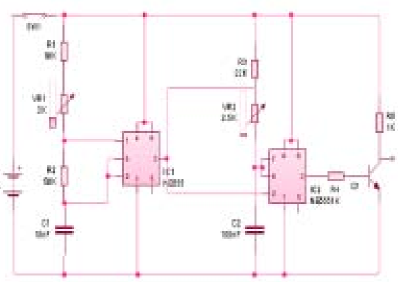




Circuit Schematic and pulse width simulation shown above.

Astable Components	Monostable Components
IC1 555	IC2 555
R1 270K	R3 10K
R2 10K	VR1 10K
C1 100nF	C2 100nF

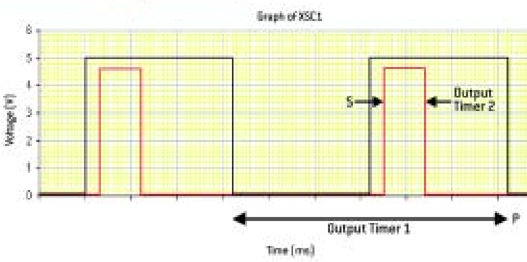
Below is a more sophisticated circuit with fine tuning for both IC1 and IC2 and transistor output if you feel adventurous. This of course needs to be duplicated for each servo.



The Brief

I set the brief to trainees to develop a device / outcome that explored the mechanical and control potential of servos. The assignment sought to draw out both mechanical and electronic skills in a sophisticated way using a maximum of two servos per project. Trainees were asked to explore the way simple but elegantly constructed mechanisms can be powered and controlled by the action of servos. Furthermore that the servos were controlled in the final presentation by a PIC or timer controlled circuit. Early experiments needed to be hard wired and or use radio TX / RX gear really to test the idea work before committing to circuit and software design.

Note the black regular '**carrier**' pulse and the modified or **interpolated** signal from the second timer in red. Timer 1 carrier pulse cycle is fixed at 20ms shown by arrow **P** and the signal pulse from Timer 2 is varied from 0.75-2.25ms shown by arrow **S**.

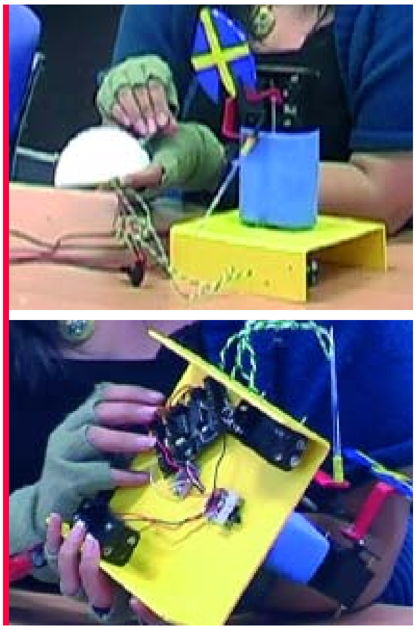


In practice most trainees developed the control and software either before completing the hardware or alongside it thus keeping two strands of work running when there were hold-ups. Additional interest could have been gained by achieving a truly wireless near range infra red (IR) control product, as IR is also an option as the signal can be sent wirelessly from a PIC or a 555 source.

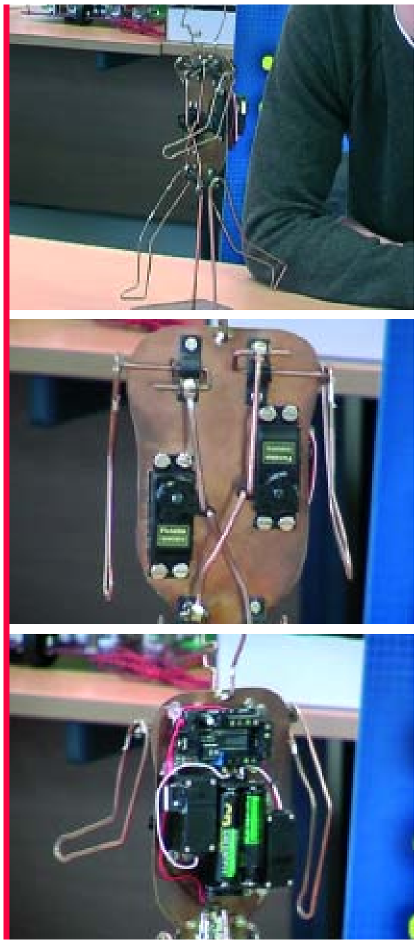
A typical engineered realisation might, for example, be an 'origamibot' or a deployable dish receiver or a humourous product dispenser.

Within the course, assessment and evaluation was based on an investigative journal, design portfolio, technical modelling and the final engineered outcome.

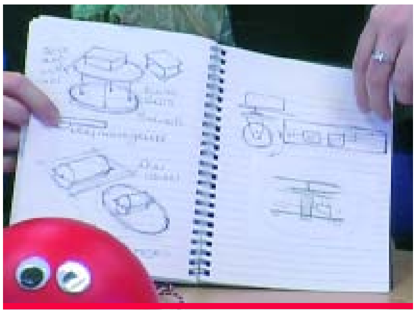
Helen's Bug Bumper car



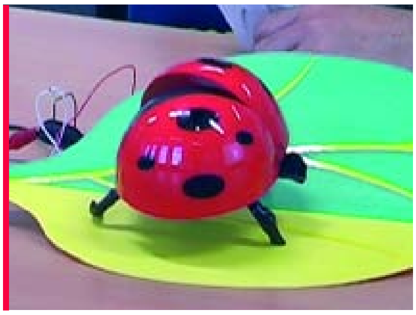
Nabila's 'Warrior'



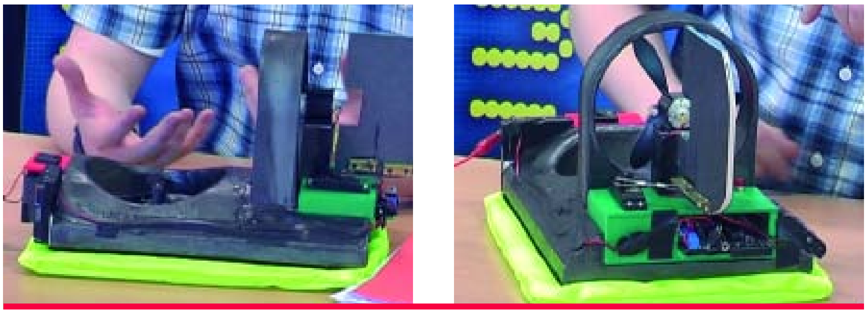
Mark's animated man showing front and rear details



Excellent opportunity to develop journal based work



Rajiv's Lady Bug – wings-a-flutter



Gareth's fully functional hovercraft

Servos with a *Smile* –

My thanks to all the Nottingham Trent University trainees and the staff and particular thanks to Allen Bower who helped with the project throughout. In practice this was an excellent project that stretched the trainees and the tutors! At a moderately low cost some intensely interesting, exotic and diverse outcomes were developed. You will certainly get closer to tidier, more elegant and sophisticated powered engineered products.