Iona Prep Robotics

Controlling a Parallax standard servo [1]

Below is a picture of a Parallax Standard Servo. The plug is used to connect the servo to a power source and a signal line (in this case both will be on the Activity Board).

1. Check that the jumper to the left of the 3 pin header labeled P12 is in the 5V position (unless your instructor tells you otherwise).
2. When you are using a servo you need to connect external power to the board. The USB port does not supply enough power.

CAUTION: NEVER move that jumper while power is applied. If you need to move the jumper you must first disconnect the battery or power supply and also disconnect the USB cable. This is very important. Moving the jumper while power is applied can cause damage to the Activity Board.

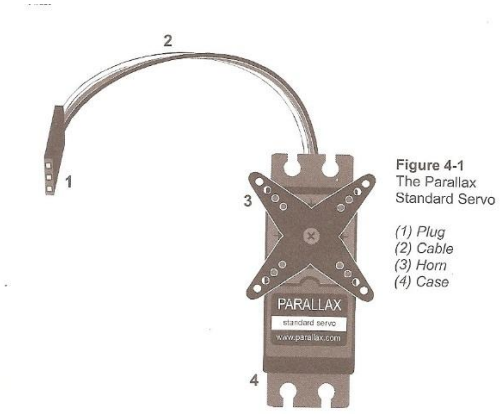
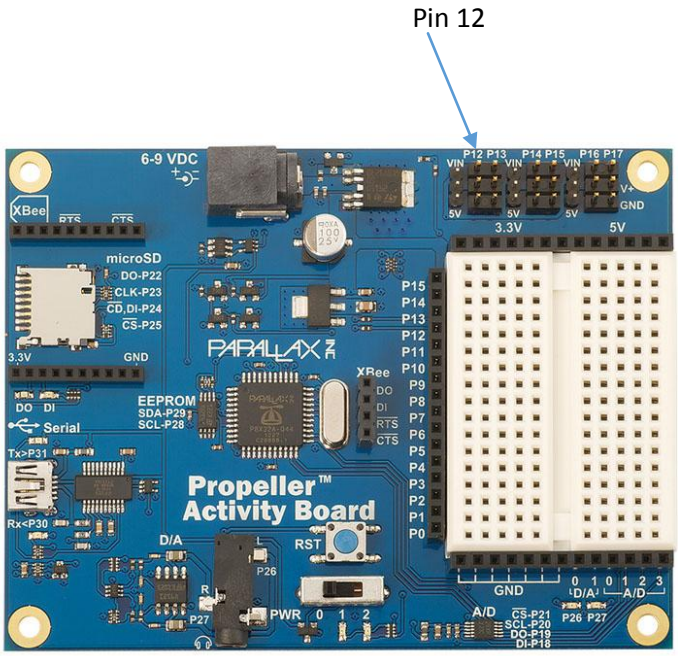


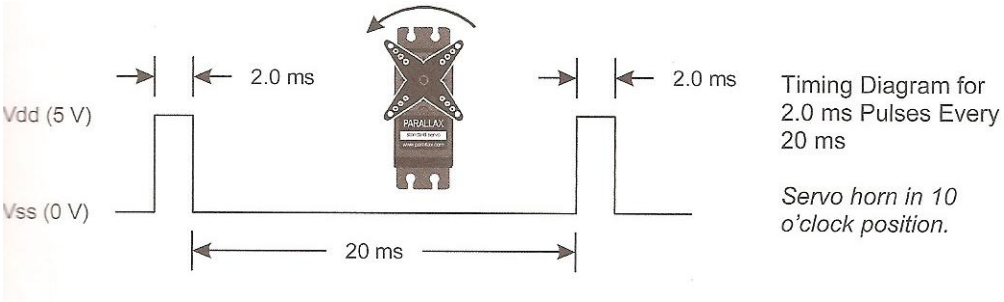
Figure 4-1
The Parallax Standard Servo

(1) Plug
(2) Cable
(3) Horn
(4) Case

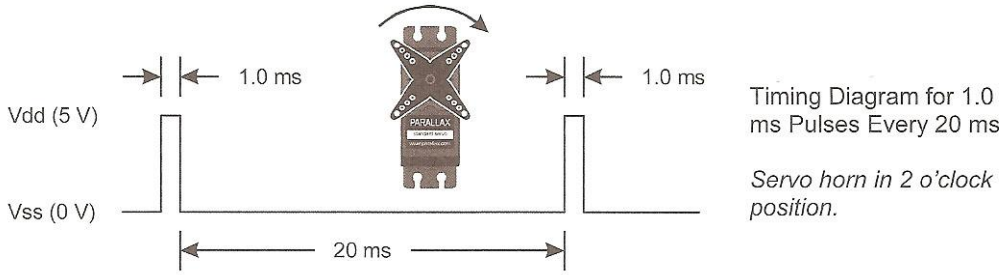


3. When you are sure the jumper is in the proper place, then plug the servo into the 3-pin header labeled P12 with the black wire closest to the white breadboard and the white wire near the top of the Activity Board.
4. A servo is controlled by very brief high pulses. Those pulses are sent over and over again every 20 ms. The high pulses last anywhere between 1 and 2 ms. The duration of the pulses determines where the servo will stop. The number of pulses determines how long the servo will hold its position. Below are pictures of the servo in three different positions:

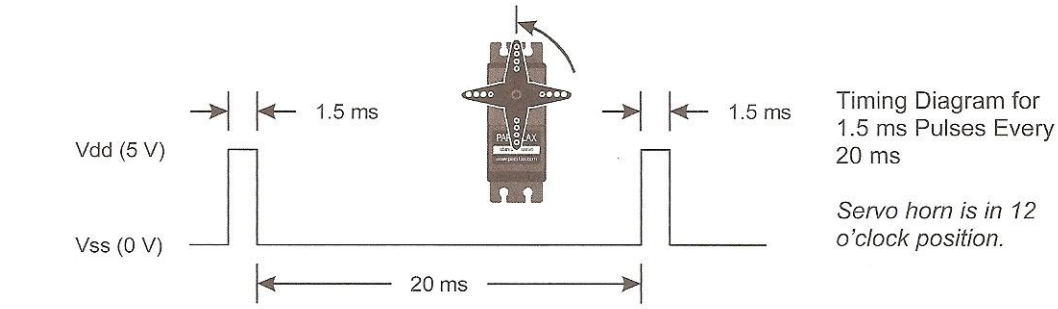
5. The following figures are called a timing diagrams. They show a picture of the high and low signals and how long they last. It does not show how many pulses are sent, it just shows the relative timing. In the first case each high pulse lasts for 2.0 ms. Each pulse is separated by a 20 ms delay when the signal is low.



The next diagram shows a 1.0 ms pulse separated from the next by 20 ms.



And the next shows a 1.5 ms pulse separated from the next by 20 ms.



6. To send out the pulses we will use two functions which are in the Simpletools library, pause and pulse_out. The pause() function takes one parameter, an integer for the duration in ms. The pulse_out() function takes two parameters, an integer for the pin and an integer for the duration in microseconds.

7. Now in order to get the horn to move to a position we need to send several pulses. Assume 150 pulses are necessary. Write a program to move the horn to its 2 o'clock position.
8. Now write a program which will move to the 2 o'clock position, hold there for 3 seconds, then move to the 10 o'clock position, hold for 3 seconds and finally move to the 12 o'clock position.

[1] Virtually all the hardware used in this course is from Parallax, Inc. The illustrations on these pages are from What is a Microcontroller, published by Parallax.

F.Y.I. This is how it would look if you programmed it in C.

```
/**
* This is the main Move Standard Servo program file.
Move servo on pin 12 to 2 o'clock
Then pause for 3 seconds
Move it to 10 o'clock
Then pause for 3 seconds
Move it to 12 o'clock
Stop.
*/
#include "simpletools.h"

int main(void)
{
int pin = 12;
int delay = 20;
int wait=3000;
int n;

for(n=1;n<=150;n++) //First go to 10 o'clock
{
pulse_out(pin, 2000);
pause(delay);
}
pause(wait);
for(n=1;n<=150;n++) //now go to 2 o'clock
{
pulse_out(pin, 1000);
pause(delay);
}
pause(wait);
for(n=1;n<=150;n++) //now go to 12 o'clock
{
pulse_out(pin, 1500);
pause(delay);
}
```

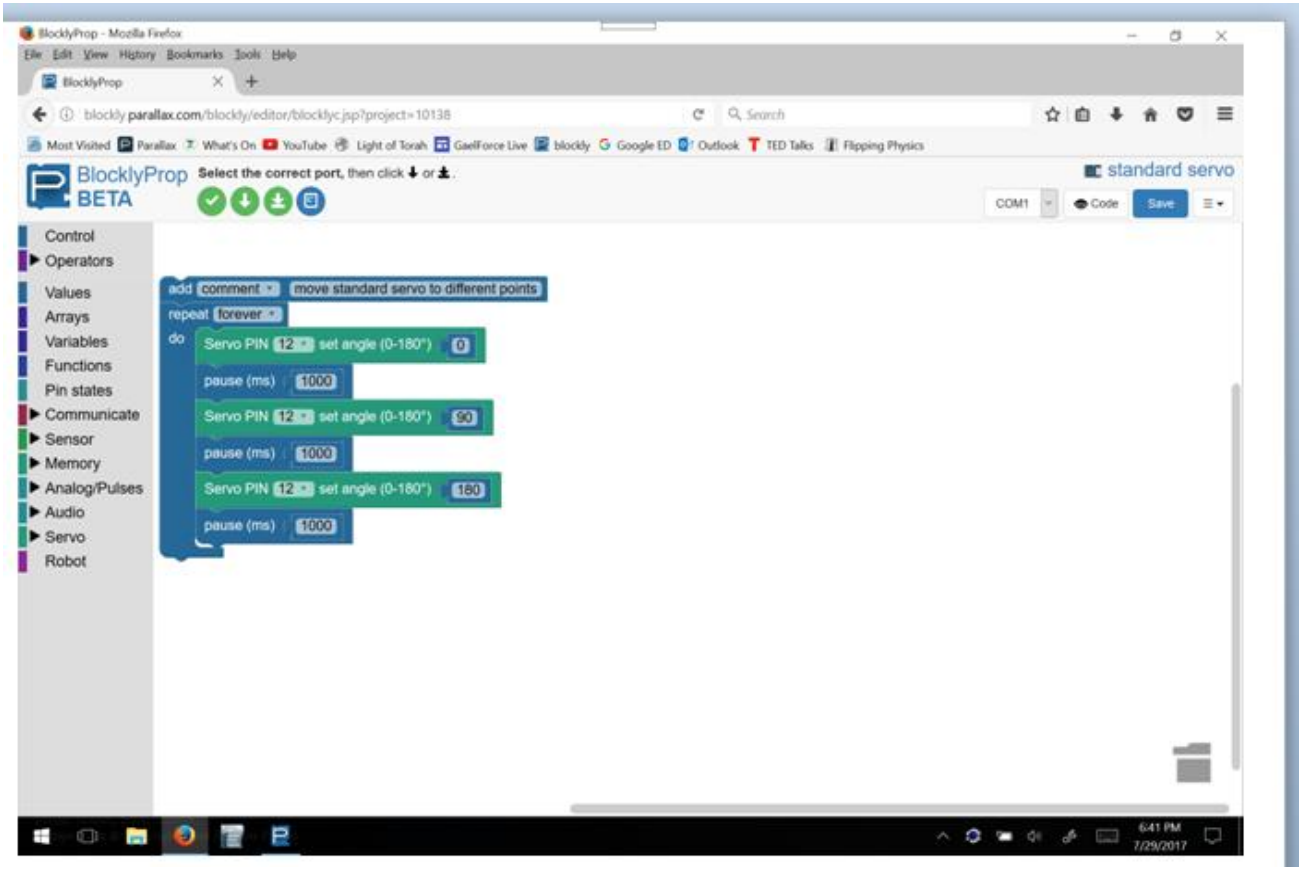
44. return 0;
45. }

Now DON'T PANIC! BlocklyProp makes it MUCH easier!

Go to this link and check out the explanation of the STANDARD SERVO.

http://learn.parallax.com/support/reference/propeller-blocklyprop-block-reference/servo

The graphic below shows how that same program looks in BlocklyProp.



That is MUCH easier. It works because someone wrote a C library to handle the messy details.