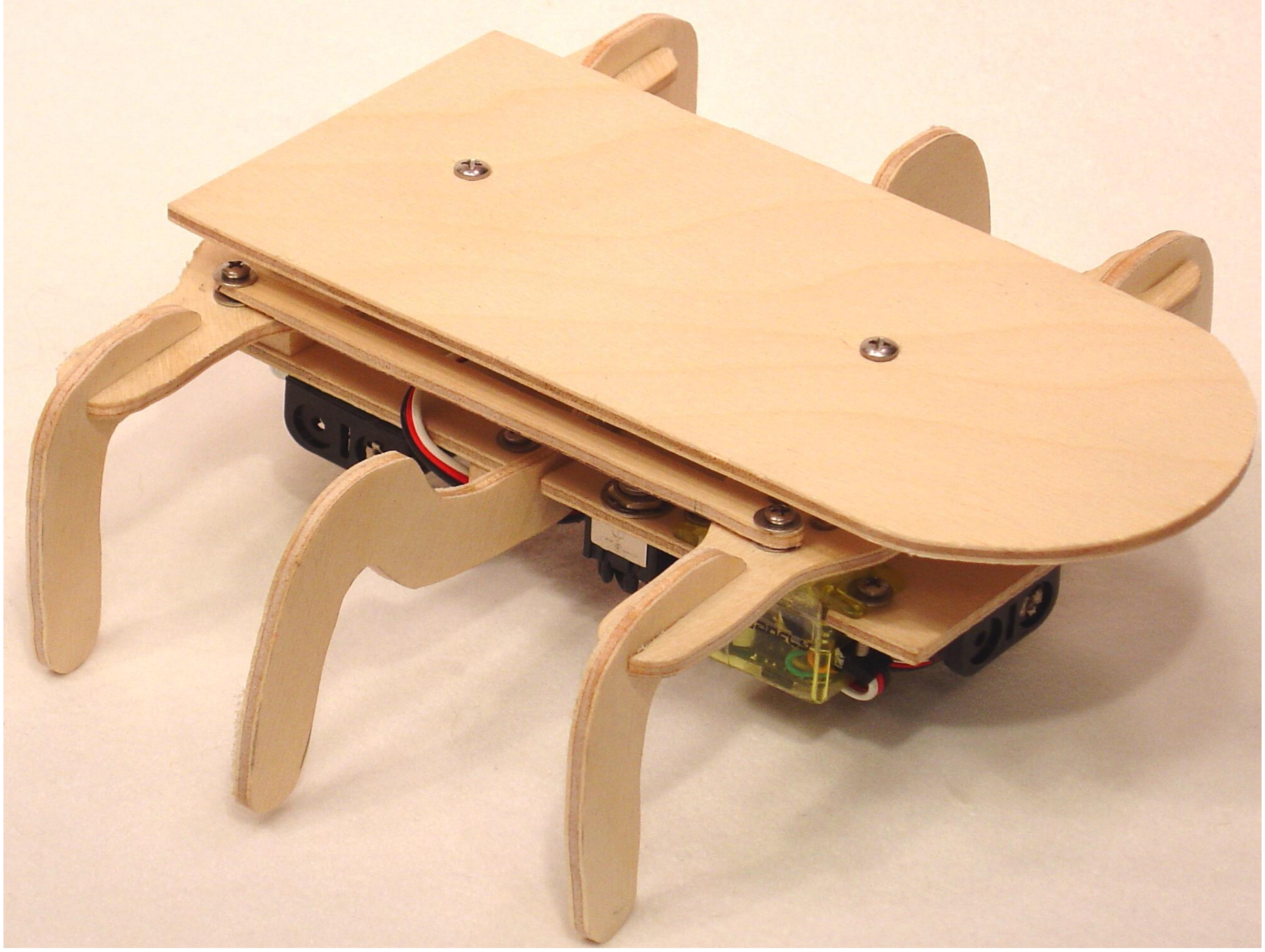
Build a Mini Servo Walker
as seen in
March/April/
June 2005 of SERVO Magazine

Pick up an issue at
http://www.servomagazine.com



Kronos Robotics
and Electronics

www.KronosRobotics.com

We've built the walker and added the power source and brain. Now it's time to place the walker in motion. In part 2 you were encouraged to play with the Perseus, so you should be familiar with programming the brain board.

Calibration

Step 1

Remove any of the screws you used to attach the servo arms (horn) to the servo. Remove the arms from the servos. This includes the servo arm from the center servo. Note that on the center servo I never did attach the servo arm screw, as the Perseus carrier makes it near impossible to insert the screw.

Rout the servo connectors up through the 5/16" hole and attach the servo connectors as follows:

- Connect Servo 0 (Left Servo) to the first servo header marked 0 on the PCB

- Connect Servo 1 (Right Servo) to the second header marked 1 on the PCB

- Connect Servo 3 (Center Servo) to the third header marked 2 on the PCB

Make sure the white (Signal) lead on the servos is facing the right edge of the board as shown in Figure 1.

Insert the EZ232 board into the 5 pin program header as shown in Figure 1. Attach your PC cable and program the Perseus with Program 1.

If you installed a switch for your walker power, make sure it does not short out against the bottom of the EZ232 board.

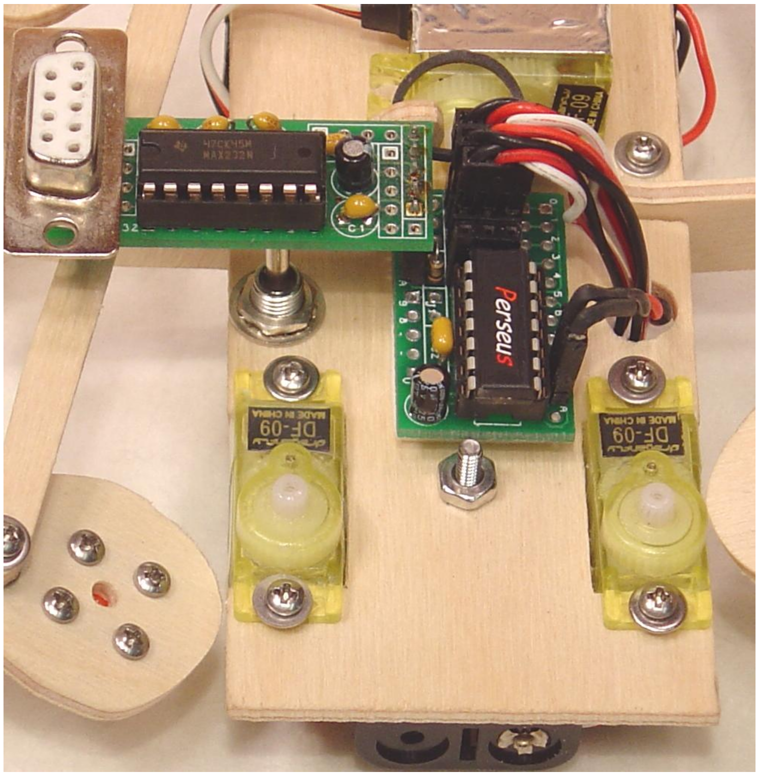


Figure 1

```
Perseus
'walkercal.txt
'
'This program will center the walker servos so
you
' Can mount all the servo arms.

   'Set ports 0-2 as output
   configio 0,1,2



loop:
     servo 0,150
     servo 1,150
     servo 2,150
     pause 15
     goto loop
```

Program 1

Once programmed, all three servos should move to the center position.

Step 2

Connect the upper leg and servo arm to the two front servos so that they are close to the center position. Note that it may not be possible to get the leg exactly centered. We will fine tune the legs in the next step.

Attach the small red arm to the center servo as close to the center (Pointing up) as possible.

Step 3

To center the legs we will load up a program that will allow you to tweak each leg.

Load Program 2 into the Perseus. The legs should assume the position that we set in step 1.

Start with the left servo and change the value of Lservocenter. Move it in increments of 5 in one direction or the other until the leg is centered exactly.

Do the same with the right servo and change the value of Rservocenter. Move it in increments of 5 in one direction or the other until the leg is centered exactly.

On the Center servo adjust the Cservocenter until it points straight up dead center.

Make a note of each servo setting. We will use these values in Program 3.

Your mini walker is now calibrated.

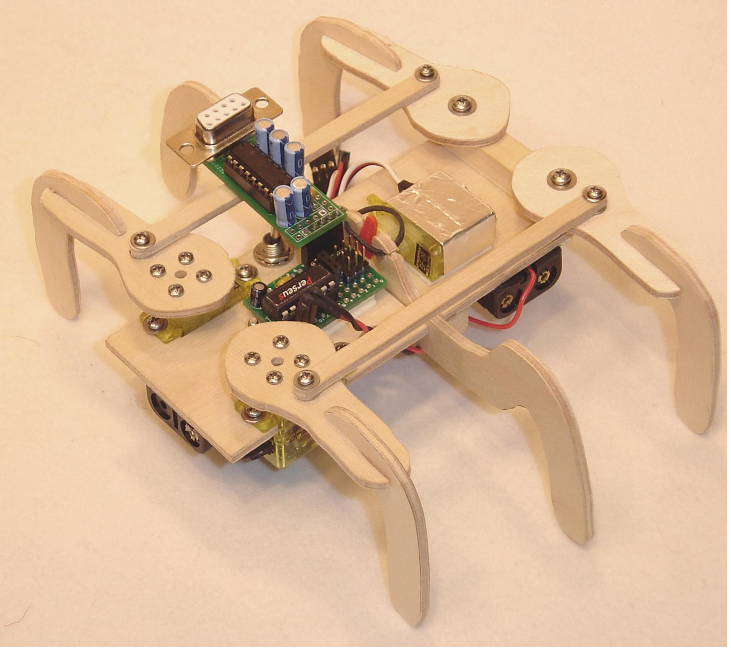


Figure 2

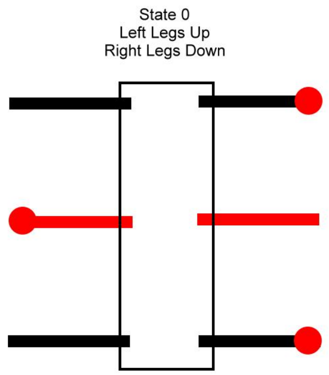
```
Perseus
'Mini servo walker calibration program 2

  dim state,cmd,ircmd,irdevice,botcmd,tim
  configio 0,1,2

'Calibrate servos here
  const Rservocenter 150
  const Lservocenter 150
  const Cservocenter 150


loop:
    servo 0,Lservocenter
    servo 1,Rservocenter
    servo 2,Cservocenter
    pause 15
    goto loop
```

Program 2

Six leg walker gates

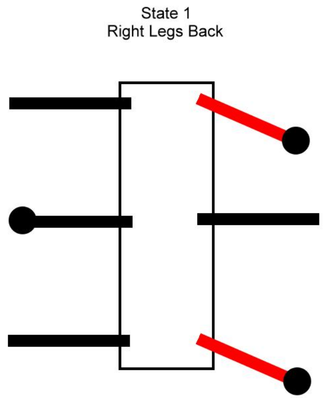
In order for a six leg walker to move we must control the legs in a particular pattern or gate. To control the patterns for each particular movement we will use a state machine. Lets look a the 6 states needed to move the walker forward in this particular gate.

In State 0 the center left leg is forced down. This forces the left front and rear legs off the ground.



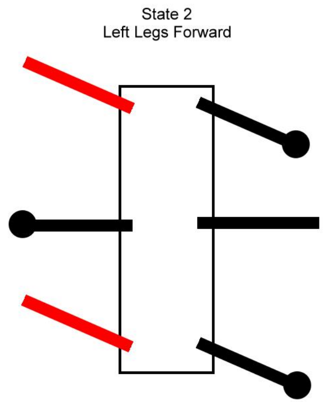
State 0

In State 1 the right front and rear legs are moved to the rear. Since these two legs are on the ground the walker's right side will move forward.



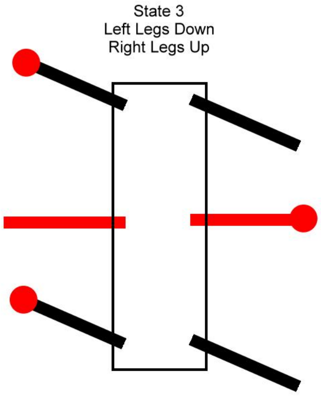
State 1

In State 2 the left front and rear legs are moved forward. They are off the ground, so it is just a positional movement.



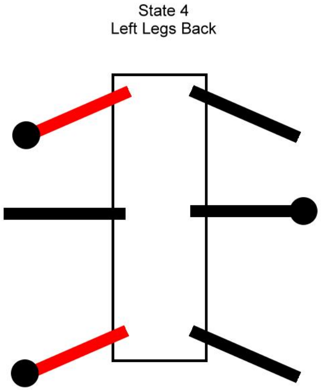
State 2

In State 3 the right center leg is forced down. This raises the right side of the walker and lowers the left side. Note that lowering the right center leg raises the left center leg.



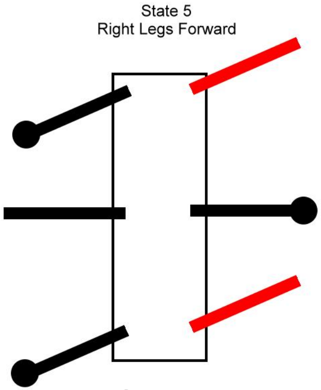
State 3

In State 4 the left front and rear legs are moved backward. Since they are on the ground the left side of the walker moves forward.



State 4

In State 5 the right front and rear legs are moved forward. They are off the ground, so this is just a positional move. At this point the walker gate has completed and the state machine starts the State 0 movements as long as we are moving forward.



State 5

There are a few variations you can make to this forward movement gate. One variation has you move the left and right legs at the same time once they are in position. This will almost double the speed of the walker. The other variation is a leapfrog style that looks something like a breast stroke.

You may have noticed that at any one time we have at least 3 legs on the ground at one time. This forms a tripod that makes this type of walker quite stable. On more complicated walkers where we have 2 or 3 servos attached to each leg we have a bit more freedom of movement. This allows for more complex gates but the priciples are still the same.

Full Walker Control

In order to use the next program you need to add a small IR module.

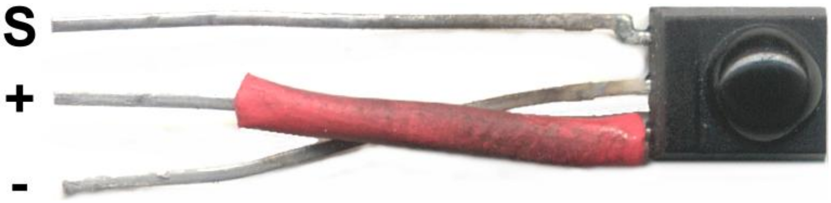


Figure 3

In order to plug the IR module into our brain board you will need to bend one of the leads as shown in figure 3. This swaps the position of the power leads so they match that of the board. Use a small piece of heat shrink to keep the leads from shorting.

Figure 4 shows how to plug the IR module into the brain board. All the other connectors have been removed only for clarity.

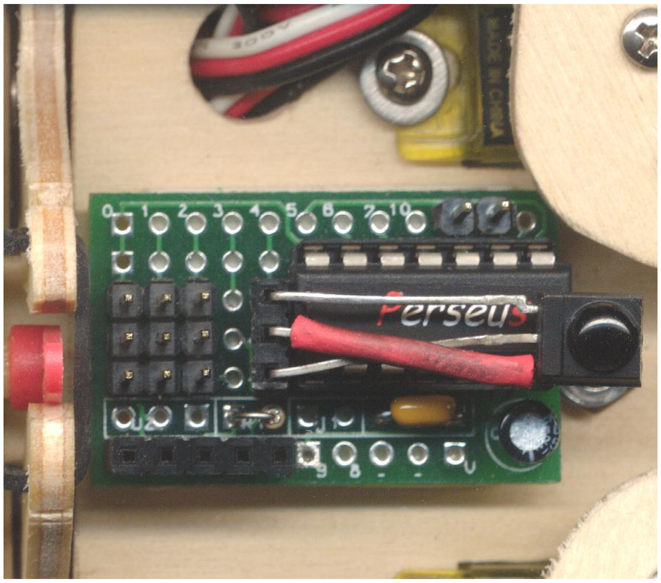


Figure 4

Program 3 will allow you to use a SONY or universal remote to control your walker. The Perseus has a command called irin. This command reads the pulses received from the IR module and translates it into data that can be used by the program.

Program 3 can be broken down into 4 sections.

Setup
Here we set up the three IO ports and all the calibration constants.
Take the three values you came up with when you calibrated the walker and plug them into the constants Rservocenter, Lservocenter, and Cservocenter.

Main Loop
In this section we look at the value returned by the irin command. We use the lookdown and branch command to jump to the appropriate state machine handler for the key hit on the remote.

State Machine
This is where the real work gets done. In order to maintain a gate we have to move the legs in a particular pattern. Using the variable state to keep track of the current state we use a lookup and branch command to form a state machine. This will cause a certain sequence of servo movements necessary to maintain the gate.

Servo Processor
Here the program moves the servos into position for the appropriate gate state. Since some states are faster than others there is a tim variable used to cycle the servos a certain number of times before control is sent back to the state machine.

Once programmed, you should be able to move the walker by hitting the following commands on the remote:

CH + = walker forward
CH - = walker backward
Vol + = walker spin right
Vol - = walker spin left

For an alternate forward gate hit the play button on the remote.

Program 3

```
Perseus

'Mini Walker IR control program.
'This Program uses a SONY IR remote to control the movement of the walker

  'Variables
  dim x
  dim state,cmd,ircmd,irdevice,botcmd,tim
  dim Rservopos
  dim Lservopos
  dim Cservopos

  PULSEINTIMEOUT = 230 'Set timeout for irin command

  'Set ports 0,1, and 2 for output.  These are the three servos
  configio 0,1,2

'Define state commands constants
  const RESET 0
  const LUP 1
  const RBAK 2
  const LFWD 3
  const RUP 4
  const LBAK 5
  const RFWD 6
  const CENTER 7
  const SPINR 8
  const SPINL 9

'Define dealy times used by servo position routines
  const shorttime 5
  const longtime 15

'Calibrate servos here
  const Rservocenter 140
  const Rservoback Rservocenter - 40
  const Rservofwd Rservocenter + 40

  const Lservocenter 160
  const Lservoback Lservocenter + 40
  const Lservofwd Lservocenter - 40

  const Cservocenter 140
```

```
   const CservoLup Cservocenter - 60
   const CservoRup Cservocenter + 60

'---------------------------------------------------------
'Main Program Loop
'---------------------------------------------------------
mainloop:

   irin 4,ircmd,irdevice

   lookdown botcmd,ircmd,0,16,18,19,17,26
   branch botcmd,botstop,botfwd,botright,botleft,botback,botquick

botstop:
   cmd = 0
   Rservopos = Rservocenter
   Lservopos = Lservocenter
   Cservopos = Cservocenter
   tim = longtime
   goto PROCITSKIP



'-------------------------------------------------------
'State machines for all walker movement
'-------------------------------------------------------
botfwd:
'Gate Forward
   lookup state,cmd,LUP,RBAK,LFWD,RUP,LBAK,RFWD,RESET
   goto cont


botquick:
'Alt Gate Forward
   lookup state,cmd,LUP,SPINL,RUP,SPINR,RESET
   goto cont


botback:
'Gate Backward
   lookup state,cmd,RFWD,LBAK,RUP,LFWD,RBAK,LUP,RESET
   goto cont

botright:
'Spin Right
   lookup state,cmd,LUP,LFWD,RUP,RBAK,CENTER,SPINR,RESET
```

```
  goto cont

botleft:
'Spin Left
  lookup state,cmd,LUP,LBAK,RUP,RFWD,CENTER,SPINL,RESET


'Walker command state processor
cont:
'Based on state go issue servo command
  branch
state,DORESET,DOLUP,DORBAK,DOLFWD,DORUP,DOLBAK,DORFWD,DOCENTER,DOSPINR,DOS
PINL

'Reset state machine
DORESET:
  cmd = 0
  goto mainloop

'Raise Left legs off the ground
DOLUP:
  tim = shorttime
  Cservopos = CservoLup
  goto PROCIT

'Right legs to the rear
DORBAK:
  tim = longtime
  Rservopos = Rservoback
  goto PROCIT

'Left legs to the front
DOLFWD:
  tim = longtime
  Lservopos = Lservofwd
  goto PROCIT

'Raise right legs off the ground
DORUP:
  tim = shorttime
  Cservopos = CservoRup
  goto PROCIT

'Left legs to the rear
DOLBAK:
  tim = longtime
```

```
   Lservopos = Lservoback
   goto PROCIT

'Right legs to the front
DORFWD:
   tim = longtime
   Rservopos = Rservofwd
   goto PROCIT

'All 4 legs flat
DOCENTER:
   tim = shorttime
   Cservopos = Cservocenter
   goto PROCIT

'right legs forward and left legs to the rear
DOSPINR:
   tim = longtime
   Rservopos = Rservofwd
   Lservopos = Lservoback
   goto PROCIT

'Right legs to the rear and left legs forward
DOSPINL:
   tim = longtime
   Rservopos = Rservoback
   Lservopos = Lservofwd
   goto PROCIT



'------------------------------
'Servo Processor
'------------------------------
PROCIT:
   cmd = cmd + 1
PROCITSKIP:
   for x = 0 to tim
     servo 0,Lservopos
     servo 1,Rservopos
     servo 2,Cservopos
     pause 15
   next
   goto mainloop
```

What's Next

The Perseus is an entry level microcontroller. Its great for getting started at a very reasonable price. Try making the walker autonomous by adding a couple sensors. The Perseus has 8 Analog to Digital ports which can be tied to many different sensors. If you want more program space you can move up to the Nemesis microcontroller and for real power the Dios will blow them all away.

The Kronos Robotics web site is located at: www.kronosrobotics.com Here you will find parts and program updates for the mini walker. This is also the home of the Athena and Dios Class of microcontrollers.