STEADY HANDS

You don't have to get complicated to get a great deal of fun from an interfacing project. Electrically this is just about as simple as you can get, however it has a very good fun to technology ratio.

DIFFICULTY: INTERMEDIATE

Steady hands is a very old game; however, with a Raspberry Pi we can give it a new twist.

The idea is that you have to guide a wire loop along a bendy wire without letting the two touch. You can make this as complex or as easy as you like by having more bends in the wire or having the loop smaller.

Materials
-Bare single core wire (2mm Dia)
e.g. metal coat hanger. If using wire it needs to be thick enough to hold its shape.
-Stranded wire (insulated)
-Block of wood (size depends on your design of 'bent coat hanger')
-Electrical tape

Tools
-Drill
-Drill bit (just smaller than the diameter of the wire)
-Solder
-Soldering iron

A - Physical pin 7 GPIO pin 4
B - Physical pin 3 GPIO pin 0
C - Physical pin 5 GPIO pin 1
D - Physical pin 6 Ground

The Construction

1. Drill holes in the wooden block for the 'bent coat hanger or wire' just smaller than the diameter of the coathanger/wire so that the wire will hold itself up. Make sure to space the two holes far enough apart to accomodate your design.

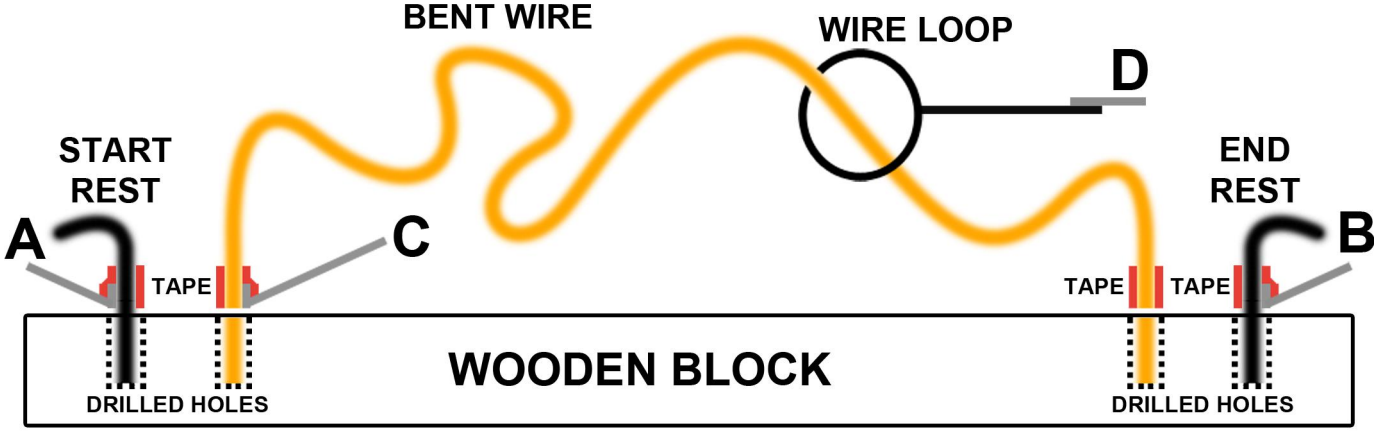
2. Make the wire loop and solder a length of wire to it. You might want to add a covering of insulation tape or, better still, self amalgamating tape over the end you hold.

3. Put the 'bent coat hanger / wire' through the wire loop and then into the wooden block.

4. Solder a length of stranded wire (insulated) to one end of the 'bent coathanger/wire'.

5. Drill two holes on either side of the 'bent coat hanger / wire', as shown in the image below for the rests.

6. Put two short lengths of coat hanger / wire, to act as rests, in these holes. These will detect when the game starts and when the wireloop reaches the end. Bend them so the loop will rest against them without shorting out on the bent wire.

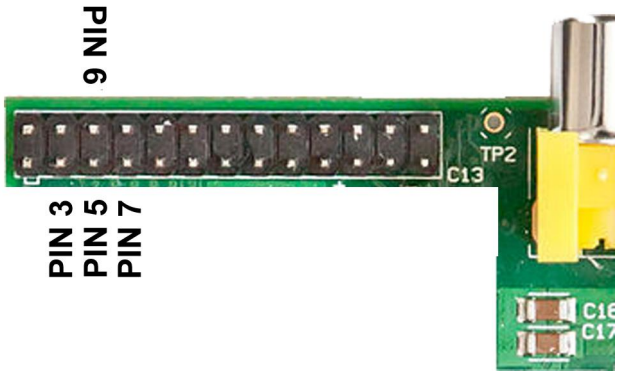


7. Solder a length of normal wire (insulated) to each end stop.

8. On each end of the 'bent coat hanger / wire' and both rests, from where they hit the block of wood, tape them with insulated electrical tape 4 cm high.

The following table and image show how each part of the 'STEADY HAND' attaches to each pin of the Pi GPIO via the 2.54mm header:

STEADY HAND	PI GPIO
WIRE LOOP	GROUND - PIN 6
BENT WIRE	GPIO 1 - PIN 5
START REST	GPIO 4 - PIN 7
END REST	GPIO 0 - PIN 3



Raspberry Pi Physical Pins

How does the GPIO work?
Basically we have three signal wires and a ground. Using GPIO 0 & 1 means that a pull up resistor is already connected on the Pi, just leaving GPIO 4 to have either an external pull up attached or activating the internal pull up resistor. I opted for the latter option.

The Program
The software was my first venture into writing in the Python language. It is quite straight forward.

First of all the three lines must be set up as inputs. They boot up as inputs anyway but it is always good practice to initialise the lines you want to use.

I used the GPIO numbers and not the physical pin numbers in the code as I strongly believe that using physical pin numbers is actually not a sensible thing to do, and not a good way to teach children. It's like the ITA (Initial Teaching Alphabet) mistake all over again.

The game is in three phases:
1) Wait until the loop is placed on the start rest.
2) Wait until the loop is removed from the start rest.
3) Time the interval from lifting it off the start rest until it reaches the end rest. While it is in this phase the Pi will monitor the bendy wire for touches.

This is then repeated forever, so a control C is needed to stop the program.

This is just the bare bones of what is possible. I always think a good way to learn anything is to extend and modify from a base. This is your base.

One extension would be to add a sound whenever the bendy wire is touched. The August issue of the MagPi showed you how to incorporate sound effects into a Python program, so take those bits and graft them into this program.

You could also keep track of the high scorer, or even have a table of high scores along with the names. You can make that permanent by writing it out to a file and reading the file when the program first starts up.

You can add penalty points into the time, say 3 seconds per point to give a single figure. On a more practical level, see if you can abort a timed run when the loop is placed back on the start loop.

There is plenty of scope for adding your own refinements. Have fun.

Continued over page...

```
# python3
# Steady hands game

import RPi.GPIO as GPIO
import time

# use BCM GPIO numbering - use anything else and you are an idiot!
GPIO.setmode(GPIO.BCM)

# set up GPIO input pins
#   (pull_up_down be PUD_OFF, PUD_UP or PUD_DOWN, default PUD_OFF)
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# GPIO 0 & 1 have hardware pull ups fitted in the Pi so don't enable them
GPIO.setup(0, GPIO.IN, pull_up_down=GPIO.PUD_OFF)
GPIO.setup(1, GPIO.IN, pull_up_down=GPIO.PUD_OFF)

print("Hi from Python :- Steady Hands game")
delay = range(0,5000)
dum = 0
start_rest = 4
end_rest = 0
wire = 1
while True:
#wait until the wand is at the start
 print("Move the loop to the start rest")
 while GPIO.input(start_rest) != 0:
   time.sleep(0.8)

#now we are at the start of the bendy wire
 print("Start when you are ready")
#wait until the loop is lifted off the wire
 while GPIO.input(start_rest) == 0:
   time.sleep(0.1)
 print("Your off")
#time the run to the other rest
 penalty = 0
 run_time = time.clock()

 while GPIO.input(end_rest) != 0:
    if GPIO.input(wire) == 0:
      penalty = penalty + 1
      print("Penalties total", penalty, " points")
      time.sleep(0.07)
 score = time.clock() - run_time + (penalty * 0.07)
  print("The  run  time  was",  score,  "seconds  with",  penalty,  "Penalty
points")
#finished a run so start again
```

Hope you give it a go. Have fun!

Article by Mike Cook