Part 5 - "What can you do with *just* Seven Segments?"

Last week I talked about some of the practical barriers to using the Raspberry Pi in the Classroom with many teachers believing they needed to upgrade their monitors to HDMI at the cost of £££ not realising that if their existing monitors support DVI all they needed was an adapter cable. Or, as is the case with many schools, they were thinking…

"Yes! ...but all our monitors are VGA!"

…all they needed was an HDMI-to-VGA Adapter!

This week I will look at how I developed a more comprehensive set of Software Challenges (and ordered a couple more Raspberry Pi's) so I could run Raspberry Pi Workshops in Schools in a Blog entitled…

"What can you do with *just* Seven Segments?"

In my role as a STEM Ambassador I had been asked by Kennet School in Thatcham to run a couple lessons for their GCSE Electronics pupils. When I worked for…

…in Thatcham about 10 years ago I was part of the Panasonic X70 Mobile Phone …

…design team, so I took along a number of old Panasonic X70 mobile phones for them to dismantle to help them understand the electronics inside. I asked if the school were doing anything with Raspberry Pi's. I was told that Mel Poyda, Head of ICT was thinking about buying some. It just so happened that I used to work with Mel at *3Net*, but hadn't realised she had gone into teaching! I discussed my ideas with her and we decided to run a pilot after-school Raspberry Pi Workshop for a group of her Year 8 pupils. I ordered 2 more Raspberry Pi's and decided their first software task would be to make the Seven Segment Display act as a Dice, a…

The pupils had never written any software before but were keen to learn! I didn't explain any of the syntax rules of Python, I just said *"look at the starting software that counts 1,2,3 on the Seven Segment Display* (See Blog Part 3) *and try and work out how by copying and modifying it you can make it count to 4."*
…*"Now make it count to 5."*
…*"Now make it count to 6!"*

I then gave them this example Python program to generate **Random Numbers**…

```
########################################################################
# Seven Segments of Pi - Seven_Segments_Random.py Random Number Generation #
########################################################################
# Description:-                                                        #
# When PushButton is pressed                                          #
# a Random Number between 1 - 6 is printed on the *Python Shell* Window #
# Example software for copying into PiDice program, etc                #
########################################################################

import random           #random package allows random number generation in the software
import time             #time package allows programmable delays in the software
import RPi.GPIO as GPIO  #RPi.GPIO package allows control of GPIO by software
GPIO.setmode(GPIO.BOARD) #Sets RPi.GPIO package to number GPIO by their Raspberry Pi Connector pin number

GPIO.setup(7, GPIO.IN)   #GPIO 7 is input from Push Button Switch

poll = .1  # Define Constant for PushButton 'poll' rate

# Start of Random Number Generator Program Execution
print "Press PushButton to Start"
while True :
    PushButton = GPIO.input(7)          # Check PushButton input
    if PushButton == False:             # if PushButton has not been pressed
        time.sleep(poll)                # wait 0.1 of a second before checking again
    else:                               # else PushButton has been pressed, so
        x = random.randrange(1,7)       # get a random number between 1 & 6
        print x                         # print this random number to Raspberry Pi *Python Shell* Window
```

…in which I had highlighted in yellow the lines they would need to add to their program to generate the random numbers needed for a **PiDice**.

The Workshop went well! Within 2 hours the 9 pupils, working in 3 groups of 3, each had a working **PiDice**!

At the next workshop their second software task was to make the Seven Segment Display act as a StopWatch, a…
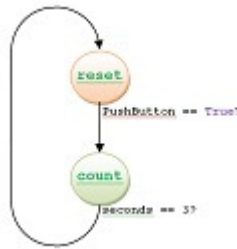

This introduced **Sound Effects**…

(which worked after I resolved a small technical glitch which I will talk about in a later Blog entitled **"Doh! There's no sound!"**)…

…and *State* Machines…

…to control the flow of the program.

This task was a bit more demanding and took two Workshops to complete!

From the *PiDice* they now knew how to write Python for:-
- GPIO as Inputs
- GPIO as Outputs
- Time Delays
- Random Numbers

And from the *PiStopWatch* they now knew how to write Python for:-
- Sound Effects
- State Machines

They now had the software skills needed to write their first computer game…

Next week I will describe how I made a PCB for the Seven Segment Display creating the **"Seven Segments of Pi" –** *possibly the simplest games console in the world!* And showed the pupils…

"300ºC…Soldering is fun!"

If you are interested in learning more about the Seven Segments of Pi visit my Web Site www.SevenSegmentsOfPi.com or watch the "Seven Segments of Pi" YouTube Video. You can also watch Carrie Anne Philbin playing "Figure Eight My Pi" at the CamJam, courtesy of Alex at RasPi TV.

Nevil Hunt
Innovations in Education