

# The Software Build and Test

I want this page to contain the Software for the Clock, and for some test patterns, for now here is the code for the clock (remember, this code is for Common Anode displays)

This code originated from the following website (although I am pretty sure that is not the original source)

<http://raspi.tv/2015/how-to-drive-a-7-segment-display-directly-on-raspberry-pi-in-python>

This code was then modified by myself and a couple of friends (thanks to Malc and Chewie) to account for a couple of changes, and these are:

1. My 7 Segment LED is Common Anode, the original code is for common cathode.
2. My 7 Segment LED has a time separator and full stops (so is 8 segments) the original does not.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

# GPIO pins used to display 7 segments
gpioSegments = (5,6,13,19,26,21,20,16)

for segment in gpioSegments:
    GPIO.setup(segment, GPIO.OUT)
    GPIO.output(segment, 1)

# GPIO pins used to select digit 0-3
gpioDigits = (18,23,24,25)

for digit in gpioDigits:
    GPIO.setup(digit, GPIO.OUT)
    GPIO.output(digit, 0)

numbers = {'0':(0,0,1,0,1,0,0,0),
    '1':(1,1,1,0,1,0,1,1),
    '2':(0,0,1,1,0,0,1,0),
    '3':(1,0,1,0,0,0,1,0),
    '4':(1,1,1,0,0,0,0,1),
    '5':(1,0,1,0,0,1,0,0),
    '6':(0,0,1,0,0,1,0,0),
    '7':(1,1,1,0,1,0,1,0),
    '8':(0,0,1,0,0,0,0,0),
    '9':(1,0,1,0,0,0,0,0),
    ' ': (0,0,0,0,0,0,0,0)}

try:
    while True:
        n = time.ctime()[11:13]+time.ctime()[14:16]
        s = str(n).rjust(4)
        for digit in range(4):
            for loop in range(0,8):
                GPIO.output(gpioSegments[loop], numbers[s[digit]][loop])
                #if (int(time.ctime()[18:19])%2 == 0) and (digit == 2):
                #    GPIO.output(23, 0)
                #else:
                #    GPIO.output(23, 1)
            GPIO.output(gpioDigits[digit], 1)
            time.sleep(0.005)
            GPIO.output(gpioDigits[digit], 0)
finally:
    GPIO.cleanup()
```

## Here is the orinial code, unedited

```
# code modified, tweaked and tailored from code by bertwert
# on RPi forum thread topic 91796
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

# GPIO ports for the 7seg pins
segments = (11,4,23,8,7,10,18,25)
# 7seg_segment_pins (11,7,4,2,1,10,5,3) + 100R inline

for segment in segments:
    GPIO.setup(segment, GPIO.OUT)
    GPIO.output(segment, 0)

# GPIO ports for the digit 0-3 pins
digits = (22,27,17,24)
# 7seg_digit_pins (12,9,8,6) digits 0-3 respectively

for digit in digits:
    GPIO.setup(digit, GPIO.OUT)
    GPIO.output(digit, 1)

num = {' ': (0,0,0,0,0,0,0),
      '0': (1,1,1,1,1,1,0),
      '1': (0,1,1,0,0,0,0),
      '2': (1,1,0,1,1,0,1),
      '3': (1,1,1,1,0,0,1),
      '4': (0,1,1,0,0,1,1),
      '5': (1,0,1,1,0,1,1),
      '6': (1,0,1,1,1,1,1),
      '7': (1,1,1,0,0,0,0),
      '8': (1,1,1,1,1,1,1),
      '9': (1,1,1,1,0,1,1)}

try:
    while True:
        n = time.ctime()[11:13]+time.ctime()[14:16]
        s = str(n).rjust(4)
        for digit in range(4):
            for loop in range(0,7):
                GPIO.output(segments[loop], num[s[digit]][loop])
                if (int(time.ctime()[18:19])%2 == 0) and (digit == 1):
                    GPIO.output(25, 1)
            else:
                GPIO.output(25, 0)
        GPIO.output(digits[digit], 0)
        time.sleep(0.001)
        GPIO.output(digits[digit], 1)
finally:
    GPIO.cleanup()
```

From:  
<http://cameraangle.co.uk/> - WalkerWiki - [wiki.alanwalker.uk](http://wiki.alanwalker.uk)

Permanent link:  
[http://cameraangle.co.uk/doku.php?id=the\\_software\\_build\\_and\\_test&rev=1485104549](http://cameraangle.co.uk/doku.php?id=the_software_build_and_test&rev=1485104549)

Last update: **2023/03/09 22:35**

