



The countdown system description. 4/21/24

The purpose is to display the date, time, the departure times of the trains scheduled for the day and the status of the next train. If and when the train will be boarding and the minutes until the train will depart.

The program runs on a Raspberry Pi Zero.

The train times come from an EXCEL file derived from the EXCEL workbook used to generate the Dispatcher's Reports. The worksheet in this workbook named Trains contains the schedule for the operating season.

The latest version will play audio announcements which can be fed to the mixer in the yard office. To access the audio, a Pi-2 or Pi-3 will be needed.

New Hope Valley Railway
Saturday, March 16, 2024
0649 **6:49 AM**
Testing the Theme Field
10:45 11:00 12:30 2:00 3:45
The next scheduled train is at 10:45.
The 10:45 train will board in 221 minutes.

['countdown.py', '2.7'] count03-2.png

New Hope Valley Railway
Friday, March 22, 2024
8:28 AM
There are no trains scheduled today.
The next train rides are scheduled for:
Fri 04/12: 1 Train at 10:30 AM In 21 days
Sat 04/13: OAL NO Passengers In 22 days
Sun 04/14: 3 Trains at 1:00 PM In 23 days
Wed 04/17: 1 Train at 10:30 AM In 26 days

['no_trainsx.py', '2.7'] no_trains9-5-4 QR



Things to know:

The original version was designed to run on a 3.5 inch monitor in full screen mode.

It is running on a 12" monitor with a magnification of 2.7 in Non-Full screen mode.

To run in magnified mode launch to program and add a space for the magnification argument.

If there is no argument it defaults to 1.

The program launches automatically when power is applied.

The *autoexec.sh* Linux shell script is run when power is applied. This launches the program.

Take note that the x versions of the programs run the NON-Full screen versions.

It contains the following three lines:

```
cd countdown
```

```
python countdownx.py 2.7
```

```
python no_trainsx.py 2.7
```

If there are no trains scheduled for the day, another program runs: *no_trainsx.py*.

This program displays the next four ride days:

The image shows a black notification screen for the New Hope Valley Railway. At the top left is a QR code. To its right is the railway's logo and the text "New Hope Valley Railway" in yellow. Below the logo is the date "Friday, March 22, 2024" in white. The time "8:28 AM" is displayed in large white digits. Below the time, the text "There are no trains scheduled today. The next train rides are scheduled for:" is shown in white. A list of upcoming train rides follows: "Fri 04/12: 1 Train at 10:30 AM In 21 days", "Sat 04/13: OAL NO Passengers In 22 days", "Sun 04/14: 3 Trains at 1:00 PM In 23 days", and "Wed 04/17: 1 Train at 10:30 AM In 26 days". At the bottom, in small white text, is the command: "[no_trainsx.py, '2.7'] no_trains9-5-4 QR".

Fri 04/12:	1 Train at 10:30 AM	In 21 days
Sat 04/13:	OAL NO Passengers	In 22 days
Sun 04/14:	3 Trains at 1:00 PM	In 23 days
Wed 04/17:	1 Train at 10:30 AM	In 26 days



Revisions:

4/21/24 rotate26.py

Properly aligned the text of the rotated images.

Changed the offsets for the three field lines from the integer of the length of the text times the cpl, which was an approximation, to the size of the surface which was made from the text.

The `surface.get_size()` function returns the width and height of the surface.

The `surface.get_size()[1]` functions returns the height of the surface.

Approximation:

```
screen.blit(display1ar,(int(m*(240)),int(m*leftoffset)-offa-int(m*len(display1a)*cpl)) ##### (H,V) NEXT RIDE
```

Actual

```
screen.blit(display1ar,(int(m*(240)),int(m*leftoffset)-offa-display1ar.get_size()[1])) ##### (H,V) NEXT RID
```

Changed offc, the offset for the days out from 330 to 355.

New Hope Valley Railway
 Sunday, April 21, 2024
7:37 AM

There are no trains scheduled today.
 The next train rides are scheduled for:

Sat 04/27:	OAL NO Passengers	In 6 days
Sun 05/05:	4 Trains at 9:30 AM	In 14 days
Fri 05/10:	1 Train at 10:30 AM	In 19 days
Sat 05/11:	OAL NO Passengers	In 20 days

The North Carolina Railway Museum, Inc. (NCRM) is an IRS registered, educationally focused 501(c)3 non-profit organization, incorporated in North Carolina.

Our mission is to preserve and share North Carolina's railroad history with the public. We are especially proud of our work with school and children's groups, special needs groups, senior groups, and many others who visit us from North Carolina, across the US, and beyond. We are a 100% all volunteer operation.

The NCRM is an "open-air" facility with exhibits that include a collection of historic and heritage railroad equipment dating back to the 1910s.

[rotate26x.py, '2'] rotate 26x

4/9/24 rotate25.py



The *length* and *width* variables were created to set the screen size in pygame. They were set to 960 x 528. This made the full screen version fit into a 27-inch monitor when magnification equals 2. In the countdown folder, enter *python rotate25.py 2* to run the program.

The *cpl* variable was created to change the characters per line used in the next trains display. It was set to 10. It is used to determine the length of the surface created by the text. Since it is only an approximation there is an issue of the location of some of the text that still has not been resolved.

It also created the *leftoffset* variable and set it to 510, instead of 490 to center the display in full-screen mode. The *leftoffset* variable is used in the rotate section for example:

```
screen.blit(texta,(int(m*(170)),int(m*.5*(leftoffset-len(display)*(cpl)))) #### (H,V)
```

New Hope Valley Railway
 Tuesday, April 09, 2024
10:33 AM

There are no trains scheduled today.
 The next train rides are scheduled for:

Fri 04/12:	1 Train at 10:30 AM	In 3 days
Sat 04/13:OAL	NO Passengers	In 4 days
Wed 04/17:	1 Train at 10:30 AM	In 8 days
Sat 04/20:	2 Trains at 1:30 PM	In 11 days

The North Carolina Railway Museum, Inc. (NCRM) is an IRS registered, educationally focused 501(c)3 non-profit organization, incorporated in North Carolina.

Our mission is to preserve and share North Carolina's railroad history with the public. We are especially proud of our work with school and children's groups, special needs groups, senior groups, and many others who visit us from North Carolina, across the US, and beyond. We are a 100% all volunteer operation.


The NCRM is an "open-air" facility with exhibits that include a collection of historic and heritage railroad equipment dating back to the 1910s.

[rotate25.py, 2] rotate 25

4/7/24 rotate22.py

Added text. *Need to change size of version and fix offsets for next trains.*



 **New Hope Valley Railway**
Monday, April 08, 2024
8:02 AM

There are no trains scheduled today.
The next train rides are scheduled for:

Fri 04/12:	1 Train at 10:30 AM	In 4 days
Sat 04/13:OAL	NO Passengers	In 5 days
Wed 04/17:	1 Train at 10:30 AM	In 9 days
Sat 04/20:	2 Trains at 1:30 PM	In 12 days

The North Carolina Railway Museum, Inc. (NCRM) is an IRS registered, educationally focused 501(c)3 non-profit organization, incorporated in North Carolina.

Our mission is to preserve and share North Carolina's railroad history with the public. We are especially proud of our work with school and children's groups, special needs groups, senior groups, and many others who visit us from North Carolina, across the US, and beyond. We are a 100% all volunteer operation.

The NCRM is an "open-air" facility with exhibits that include a collection of historic and heritage railroad equipment dating back to the 1910s.

[rotate22.py, '1.9'] rotate 20

The text was taken from the triangletrain.com website.
The text must be gray characters on a black background

In MS-WORD make text 21 characters per line
Copy text to DesignCad. Fix any spacing errors
Make box around characters.
Move box to new location.
Use Draw, Hatch fill with Black.
Make Text a Group
Make Copy and move to new location
Make Text Color 250,250,250
Move over black background
Zoom to full screen
Copy this to MS-PAINT
Save as info.png

Note: info is a surface from the first line below!

Program code added

```
info = pygame.image.load('info.png').convert() ##Makes surface
```



```
new_size_info = (int(m*470),int(m*470))    ## Makes full size
info = pygame.transform.scale(info,new_size_info) ##Scales the surface to full size
infor =pygame.transform.rotate(info,90)     ## Rotates Surface

screen.blit(infor,(int(410*m),1))  # Displays Surface
```

The display above was launched by *python rotate22.py 1.9*

The North Carolina Railway Museum, Inc. (NCRM) is an IRS registered, educationally focused 501(c)3 non-profit organization, incorporated in North Carolina.

Our mission is to preserve and share North Carolina's railroad history with the public. We are especially proud of our work with school and children's groups, special needs groups, senior groups, and many others who visit us from North Carolina, across the US, and beyond. We are a 100% all volunteer operation.

The NCRM is an "open-air" facility with exhibits that include a collection of historic and heritage railroad equipment dating back to the 1910s.

4/5/24 rotate20.py

To display a rotated surface on the screen:



1- A font must be defined:

```
font = pygame.font.SysFont('Calibri', int(m*20),  
True, False)
```

2- The text must first be converted to a surface:

```
display1ar = font.render(display1a,True,GRAY2)
```

3- The surface must be rotated:

```
display1ar = pygame.transform.rotate(display1ar,90)
```

4- The surface must be located in the screen:

```
screen.blit(display2ar, (int(m*(270)),int(m*490)-offa-  
int(m*len(display2a)*10))) #### (
```

In 1- the font style is defined and the size is 20

In 2-, the surface is defined, based on the text and the color is set to GRAY2

In 3-, the surface is rotated 90 degrees.



In 4-, it is positioned 270 pixels from the left edge and 490 lines down from the top less the offa and the length of the text, times 10 pixels per letter.

This version added the 4 next ride days. There is an error in the positions of the displayxbr. This is caused by the len(displayxb) function. The length was estimated as 10 pixels per letter. This is an approximation. What is needed is the exact length or size of the surface displayxbr. This was not a problem with non rotated surface because the position is the top left corner of the surface,

3/31/24 Rotate 14

This is the display from *python rotate14.py*

A digital display with a black background. At the top left is the New Hope Valley Railway logo. Below it is the date "Sunday, March 31, 2024". The time "7:44 AM" is displayed in large white font. Below the time is the text "There are no trains scheduled today. The next train rides are scheduled for:" followed by "This is a test line" in yellow. A QR code is located in the bottom left corner.


Sunday, March 31, 2024
7:44 AM
There are no trains scheduled today.
The next train rides are scheduled for:
This is a test line


This is the display from *python rotate14.py 2.1*

≡≡≡ New Hope Valley Railway ≡≡≡



≡≡≡ New Hope Valley Railway ≡≡≡

Sunday, March 31, 2024

7:46 AM

There are no trains scheduled today.
The next train rides are scheduled for:
This is a test line



To rotate the text:

- 1- Define the textx
- 2- Render the textx
- 3- Rotate the textx



4- Screen.blit the textx

For the timee display, the -100 is a fudge!

```

screen.blit(textt, (int(m*(80)), int(m*.5*(490-100-len(timee)*(10)))))) ##### (H,V)

textt = font.render(timee, True, GRAY2) ## textt is for timee
textt = pygame.transform.rotate(textt, 90)
font = pygame.font.SysFont('Calibri', int(m*20), True, False) #For other lines

textxx = 'This is a test line'
textx = font.render(textxx, True, TEAL)
textx = pygame.transform.rotate(textx, 90)

screen.blit(logor, (1, int(m*65))) # Rotated
screen.blit(qr, (int(m*10), int(m*420))) #Rotated
screen.blit(text, (int(m*(50)), int(m*.5*(490-len(Top_date)*(10)))))) ##### (H,V)
screen.blit(textb, (int(m*(150)), int(m*.5*(490-len(displayb)*(10)))))) ##### (H
screen.blit(texta, (int(m*(170)), int(m*.5*(490-len(display)*(10)))))) ##### (H,V)

screen.blit(textt, (int(m*(80)), int(m*.5*(490-100-len(timee)*(10)))))) ##### (H,V)

screen.blit(textx, (int(m*(190)), int(m*.5*(490-len(textxx)*(10)))))) ##### (H,V)

```

3/24/24

Rotate 10 began adding lines. Must do timee

3/23/24

For the rotated panel:

The screen is

3/22/24 Rotating the panel. rotate8.py

The screen size is: `pygame.display.set_mode((int(m*1800), int(m*490)), 0, 32) ## sets full screen`

1800X490 lines

With the character size, there are 48 characters per width.

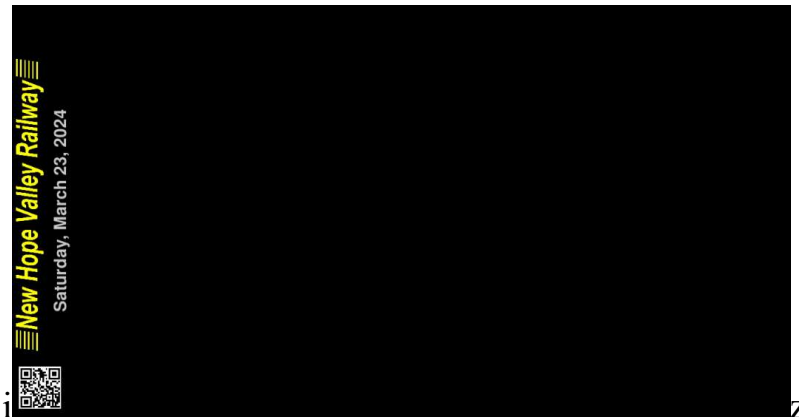
So the offset to center the text is `0.5*(490-10*len(text))`



```
Or .5*(width-Characters_per_width*len(text))
screen.blit(text, (int(m*(50)),int(m*.5*(490-len(Top_date)*(10)))) ## (H,V)
```

```
width = 490
characters_per_width = 10
```

The maximum magnification is 2.2. That fills the screen of a 16:9 monitor
 This is the display with a magnification of 1.



A system font was added `font = pygame.font.SysFont('Calibri', 20, True, False)`
 to make `text = font.render(Top_date, True, BLUE)` work.

3/22/24 Version no_trains8-5-4x.py added QR code to the display.

3/19/2024



Version *countdown8-1x.py* added variables *play1* through *play4* and plays the short audio tracks.

Playback time	Audio track name
16	<i>1.mp3</i>
11	<i>2.mp3</i>
6	<i>3.mp3</i>
2	<i>4.mp3</i>

These are initially set to *False*. As each audio track is played, they are set to True so that the tracks will play only once. When the time of departure is less than 1 minute, they are all set back to *False*.

```

if tmin[tr]-tnow == 16 and play1 == False:
    play1 = True
    os.system('omxplayer audio_tracks/1.mp3')
elif tmin[tr]-tnow == 11 and play2 == False:
    play2 = True
    os.system('omxplayer audio_tracks/2.mp3')
elif tmin[tr]-tnow == 6 and play3 == False:
    play3 = True
    os.system('omxplayer audio_tracks/3.mp3')
elif tmin[tr]-tnow == 2 and play4 == False:
    play4 = True
    os.system('omxplayer audio_tracks/4.mp3')
else:
    if tmin[tr]-tnow < 2:
        play1 = play2 = play3 = play4 = False

```

3/18/23

Version *countdown8-1x.py* added playing audio clips before the train departs. The audio clip must be greater than 60 seconds otherwise the clip will repeat. It also stops updating the clock during that time. A better solution is to not replay the clips.

The mp3 audio clips must be in the */home/pi/countdown/audio_tracks* folder. They will be played at 16, 11, 6 and 2 minutes before scheduled departures so that they do not coincide with the same audio tracks played on the train.

Playback time	Audio track name
---------------	------------------



- 16 *1_1-minute.mp3*
- 11 *2_1-minute.mp3*
- 6 *3_1-minute.mp3*
- 2 *4_1-minute.mp3*

3/16/23

Version *countdown7-3-2x.py* added military time. Variable *timeem* and *show_timeem()*



3/15/23

Version *countdown7-3-1x.py* fixes problem with *theme* display.

The next version will add audio pre-departure clips. This will have to be developed on a Pi-3.

The Pi-5 has no audio and does not support *omxplayer*.

3/12/24

The system will sort of run in windows. The format for the variable *timee* had to be changed to '%H:%M:%S' because windows did not support the *strftime* format.

```
#timee = ' '+now.strftime('%-I:%M %p')
```

```
timee = ' '+now.strftime('%H:%M:%S')
```

The files are *countdownwinx7-3.py* and *no_trainswinx9-5-3.py*

BUT: It crashes if anything else is moved on the screen.

3/9/24

For the monitor in the yard office, the magnification is 2.7.

The *autoexec.sh* file is



```
pi@raspberrypi:~ $ cat autoexec.sh  
cd countdown  
python countdownx.py 2.7  
python no_trainsx.py 2.7
```

Added magnification to countdown *cntdwn7-3x.py*.

This version does not display the theme and does not center the logo properly

Note that version *cntdwn7-3-1x.py* changed the logo file name to *nhvrr_logo.jpg* which is installed in yard office.

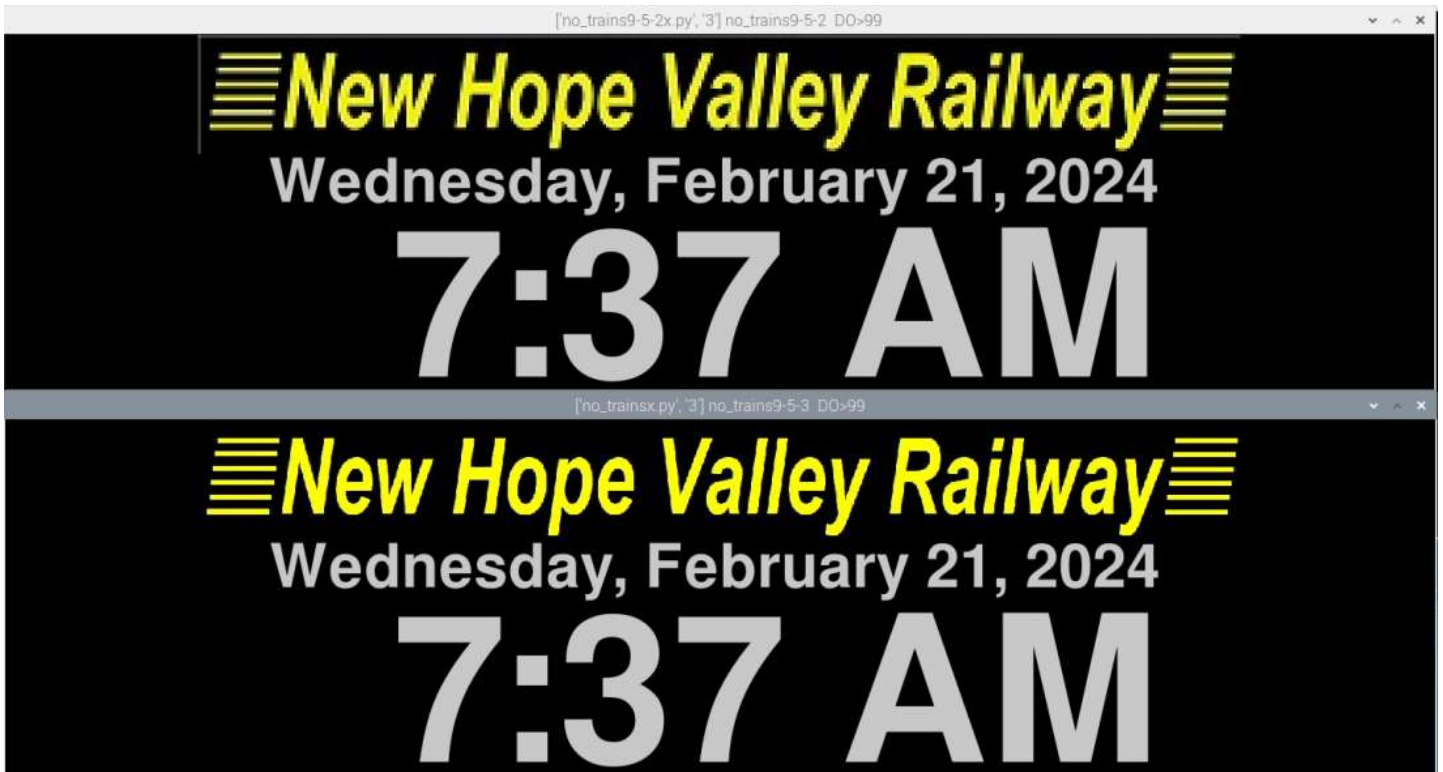
2-20-24

When the magnification was added, the graphic *nhvrr350.jpg* gets raggedy.

In version 9-5-3, the logo was changed from using the *nhvrr350.jpg* to *logo_1450x164.jpg*.

This was then reduced by a factor of .241 to fit into the 480x320 3.5" display.

$350/1450 = 0.241$



The logo_1450x164.jpg was created in DesignCad and copied to Draw where it was resized and saved as a jpg file.

```
pygame.display.set_caption(str(script)+' '+ ver) # Title of window
ball = pygame.image.load('logo_1450x164.jpg').convert()

new_size = (int(m*.241*1450),int(m*.241*164))
ball =pygame.transform.scale(ball,new_size)###
```

In version 9-5-2, the magnification variable was changed from an integer to floating point. This makes it possible to magnify the display to any desired size. It required that wherever m was used the `int(m*x)` had to be changed.

In version 9-5-1, the capability to magnify the display was added. The magnification can be changed by adding it when launching the program. To start the program, at the command line



enter *python no_trains9-5-1x.py m*. Where *M* is the magnification. If *m* is not added the default is 1, which is 480x320, for the 3.5” display. *M* must be an integer.

For example: To triple the size, at the command line, enter: *python no_trains9-5-5x.py 3*.
The function

The argument *argv* *argv*[0] is the name of the file. The first argument *argv*[1] is the first argument that is passed to the program.

In this case *argv*[1] is the magnification that is desired. Since it is formatted as text, the *int(argv*[1]) converts it to an integer.

The code added is:

```
try: # Sets m = 1 if no data is available to pass  
    m=int(argv[1]) # Magnification of the images  
except:  
    m=1
```

Added variable *m* to magnify the images, *no_trains9-5x.py*. The variable *m* must be an integer. If *m = 1*, the image size is correct for the 5.5” display ie. 480x320.

Offsets: *offa*, *offb* and *offc* etc

The best way to do this is to add the variable to the program on boot up ie *no_trains.py,m*.

Or, store the value of *m* in a text file *mag.txt* and have the program open this file if it exists and get the value.

This modification as only made for the *no_trains*, not *countdown*.



NOTE: The date format MUST be in the form *dd/mm/yyyy* and the time format must be capital *AM* and *PM*.

This can easily be changed in EXEL.

This file must be installed in the folder that contains the *countdown.py* program once a year and update if there are any changes.

The system requires a WiFi network in order to set the clock to real time.

A feature was added that will play announcements automatically to alert passengers of the status of the next train ride.

Other than the above conditions, the system runs autonomously.

Just apply power to the system. No other user interaction is required.

The system can run continuously. The startup script, *autoexec.sh*, runs the *countdown.py* program. This program searches the *schedule.csv* file for the days date. If there are no trains scheduled for the day, the program ends. The script then runs the *no_trains.py* program. This program searches the *schedule.csv* file to find the next scheduled ride day. If this an Operate-A-Loco, it display both the Operate-A-Loco and the next ride days date and starting time.

When the time = 00:00:01, whichever program is running will cause the system to reboot to load the *schedule.csv* file and start the process again.

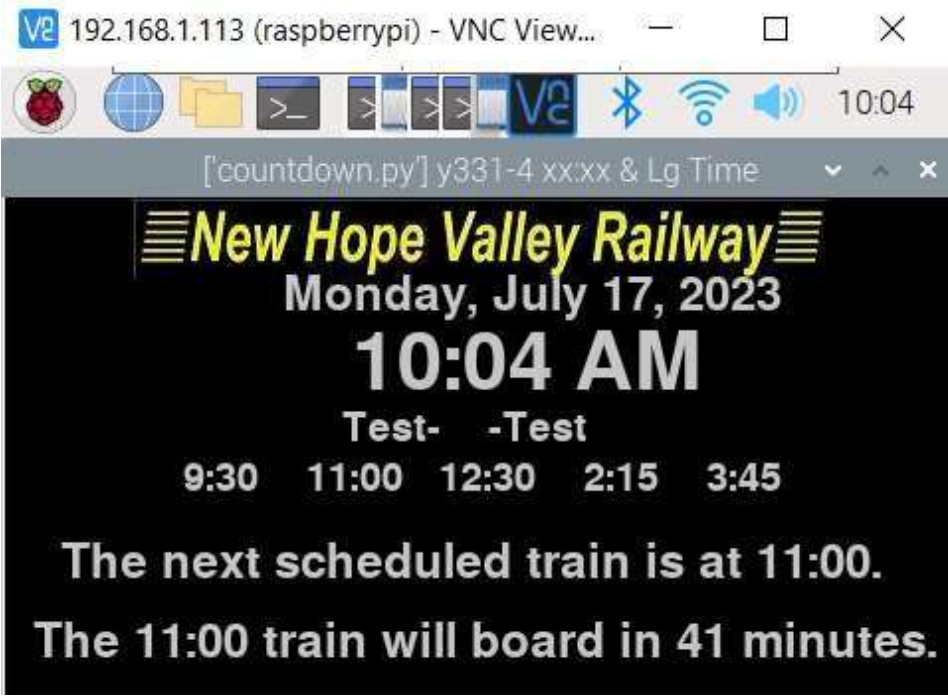
Each program will also write to the *bootlog.txt* file, the file name and the time the system was rebooted.

This is the *autoexec.sh* file:

```
cd countdown
python countdown.py
python no_trains.py
```



When the autoexec.sh Linux shell script is launched on powering up the Rasp Pi, it runs the countdown.py and the display is shown below:



If there are no trains scheduled for the day, the program crashes and the autoexec.sh script runs the no_trains.py program and the display is shown below:



New Hope Valley Railway
Friday, February 09, 2024
3:50 PM
There are no trains scheduled today.
The next train rides are scheduled for:

Wed 03/20:	1 Train at 10:30 AM	In 40 days
Fri 04/12:	1 Train at 10:30 AM	In 63 days
Sat 04/13:	OAL NO Passengers	In 64 days
Sun 04/14:	3 Trains at 1:00 PM	In 65 days

[no_trains.py] no_trains9-1-4 DO>99

At midnight the Rasp Pi is rebooted to reload the schedule and determine if there are train rides on that day.

```
pi@raspberrypi: ~/countdown3
File Edit Tabs Help
pi@raspberrypi:~ $ cd countdown3
pi@raspberrypi:~/countdown3 $ cat bootlog.txt
##### Boot Log File

Reboot from NO TRAINS Monday, May 15, 2023      8:26 AM
pi@raspberrypi:~/countdown3 $ cd countdown3
```




bootlog.txt



2023 Schedule and Ridership

Total Ridership YTD 1950

Revised 4/17

5/12/2023

Index	Date		Trains	1	2	3	4	5	Riders	Average	Theme
1	02/01/23	Wed	1	10:30 am					1	1	Photo shoot
2	03/31/23	Fri	1	11:00 am					3	3	VIP Train
3	04/01/23	Sat	5	9:30 am	11:00 am	12:30 pm	2:15 pm	3:45 pm	410	82	Hop into Spring-1
4	04/02/23	Sun	5	9:30 am	11:00 am	12:30 pm	2:15 pm	3:45 pm	707	141	Hop into Spring-2
5	04/12/23	Wed	1	10:30 am					140	140	Group ride
6	04/15/23	Sat	4	11:00 am	12:00 pm	1:00 pm	2:00 pm				Operate-A-Loco
7	04/21/23	Fri	1	10:30 am					193	193	Group ride
8	04/23/23	Sun	1	2:30 pm					121	121	Charter
9	05/06/23	Sat	4	1:00 pm	2:30 pm	4:00 pm	5:30 pm		375	94	Brew & Choo
40	05/12/23	Fri	1	10:30 am							Group ride
44	05/13/23	Sat	4	11:00 am	12:00 pm	1:00 pm	2:00 pm				Operate-A-Loco
12	05/17/23	Wed	1	10:30 am							Group ride
13	05/20/23	Sat	1	TBD							Fuel train
14	05/21/23	Sun	3	1:00 am	2:30 pm	4:00 pm					Slow Down Sunday
15	05/26/23	Fri	1	10:30 am							Group ride
16	06/03/23	Sat	4	1:00 pm	2:30 pm	4:00 pm	5:30 pm				Brew & Choo

schedule.csv

NOTE: The date format MUST be in the form *dd/mm/yyyy* and the time format must be capitol *AM* and *PM*.

This can easily be changed in EXEL.

The system displays the boarding time. The trains will board 15 minutes before their departure time. The system will dis

The system displays the boarding time. The trains will board 15 minutes before their departure time. The system will dis

NOTE: The files that end in x, such as no_trains7-5-3x.py are not FULLSCREEN versions. The line below is commented out to inhibit full screen.

```
##DISPLAYSURF = pygame.display.set_mode((480, 320), pygame.FULLSCREEN)
```

This is necessary during testing otherwise the Pi would have to be rebooted to make any changes.



no_trains2.py adds bootlog.txt

Version 323 adds bootlog.txt

Version 322 repairs the issue at the departure time and the next train boarding time.

The countdown_big.py version increases the display by a factor of 4.

The magnification variable m was added and set to 4.

The logo file nhvrr350.jpg was fuzzy when scaled up so a new logo was created to make the image sharper, nhvrr_logo.jpg.

This was too large and had to be scaled down by half in the pygame window

```
#####
```

```
image = pygame.image.load("nhvrr_logo.jpg") width
```

```
= image.get_rect().width height =
```

```
image.get_rect().height
```

```
image = pygame.transform.scale(image, (int(width*.5), int(height*.5)))
```

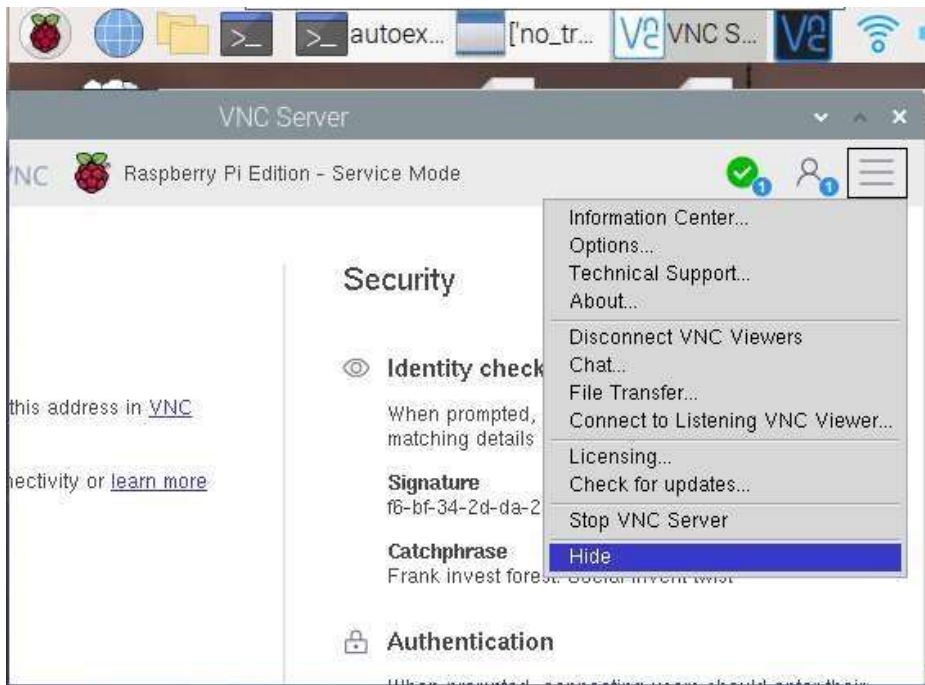
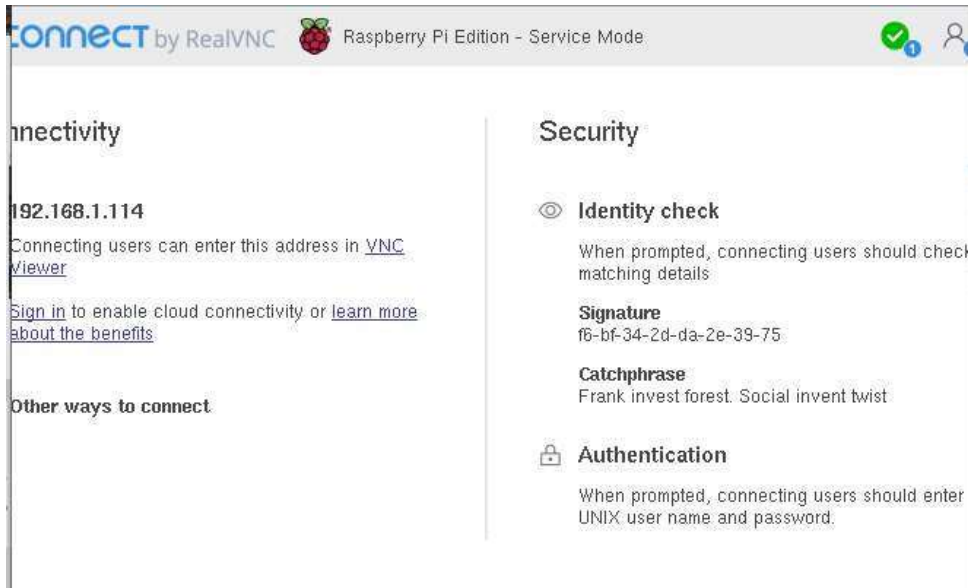
Test file in c:\Python27\cd1.py

Shows train departure times in seconds from mid night

The file cd11.py displays all trains and there difference between time now and departures

Note: To transfer files from the Raspberry Pi with the 3.5 inch tft display you must reboot the system using ./hdmi.

Otherwise, you cannot move the window to display the dropdown menu.



Then reboot the system using `./tft`

Countdown system for the Ticket Office

The count down system displays the date, the current time and the times the trains will depart .



It also displays the boarding time and the departure time of the next train leaving Bonsal

The system determines the date and time through its connection to the local area network through its Wi-Fi connection

Section 1: imports the necessary routines.

Section 2: Functions needed.:

1. `convert24()`

This converts the time in the form 9:30 AM or 9:30 PM to military time, ie 24hr time.
9:30 AM \square 0900 and 9:30 PM \square 2130

2. `convert_to_min()`

This converts the time in hh:mm:ss into minutes from midnights

Section 3: imports the data found in the 202X Trains EXEL file. The schedule work sheet is exported as a csv file. The file name is schedule.csv. The data is imported into the program as an array.

It also formats the time data to display on the monitor.

The next section searches the array for the current day's schedule and formats it for the display.

Section 4: Changes the string data from the array to seconds from midnight. Each of the train's departure times are converted to an integer which is number of seconds from midnight.

Section 5: Searches the imported array data for today's train schedule.

The next section sets up the video display using pygame

Section 6: The last section is the while loop that updates the screen.

Functions:

`Convert24()`



This function converts the time information from the schedule.csv file military time. The .csv file has the departure time information in the form of:

9:15 AM or 3:30 PM

The output of this function is 0915 and 1530.

The first part finds the colon and adds a 0 if needed.

If meridian = AM and the hour is not 12

8/3/23 The no_trains6.py program was modified to show the next schedule train ride day.



8/6/23

Since the Operate A Local ride days have no passengers, if that is the next ride day, I added the following ride day as the next ride. I also added OAL to the display.



no_trains7-1.py This version added:

The AOL line on displayd.

The number of trains and time of first train

It moved the displayc down to make room for displayd.

```
def show_displayc(message):
```

```
    font = pygame.font.Font(None,30) text =
font.render(message, 1,(200,200,200))
screen.blit(text,(30,260)) ### Hor.Vert
```

```
def show_displayd(message):
```

```
    font = pygame.font.Font(None,30) text =
font.render(message, 1,(200,200,200))
screen.blit(text,(100,240)) ### Hor.Vert
```

The following code creates the OAL displayd, in version 7-1

```
display = " There are no trains scheduled today." displayb =
' The next train rides are scheduled for' print ('Theme for
nexttrain '+stheme[nexttrain]) if stheme[nexttrain]
=='Operate-A-Loce':
    displayd = sdow[nexttrain] +' '+sdate[nexttrain]+' '+stheme[nexttrain]
nexttrain = nexttrain+1 else:
    displayd = ''
    displayc = sdow[nexttrain] +' '+sdate[nexttrain]+' ' +snot[nexttrain]+' Trains starting at '
+strainl[nexttrain]
    if snot[nexttrain] == '1': ## if there is only one train displayc = sdow[nexttrain] +'
'+sdate[nexttrain]+' ' +snot[nexttrain]+' Train at '+strainl[nexttrain]

    nexttrain = nexttrain -1 ## resets nexttrain Do not know why it is necessary!
```

8/12/23

This is the display from no_trains7-5.py showing the names for the display variables.



Top_date
timeē

display
displayb
displayd
displayc
str(script)+' '+ver

```
timee = ' '+now.strftime("%-I:%M %p") display = " There are no trains scheduled today." displayb = ' The
next train rides are scheduled for' print ("Theme for nexttrain '+stheme[nexttrain])
```

```
displayd = sdow[nexttrain] +' '+sdate[nexttrain]+' '+snot[nexttrain] +' Trains starting at '
+strain1[nexttrain] if stheme[nexttrain] ==
'Operate-A-Loco':
    displayd = sdow[nexttrain]+' '+stheme[nexttrain] +' NO Passengers' elif
snot[nexttrain] == '1':
    displayd = sdow[nexttrain]+' '+sdate[nexttrain][0:4]+' ': '+snot[nexttrain] +' Train at '
+strain1[nexttrain] else:
    displayd = sdow[nexttrain]+' '+sdate[nexttrain][0:4]+' ': '+snot[nexttrain] +' Trains starting at ' +strain1[nexttrain]
```

```
displayc = sdow[nexttrain+1] +' '+sdate[nexttrain+1][0:4]+' ' +snot[nexttrain+1]+' Trains starting at '
+strain1[nexttrain+1] if stheme[nexttrain+1] ==
'Operate-A-Loco':
    displayc = sdow[nexttrain+1]+' '+sdate[nexttrain+1][0:4] +': '+stheme[nexttrain+1] +' NO Passengers' else:
    displayc = sdow[nexttrain+1] +' '+sdate[nexttrain+1][0:4]+' ': '+snot[nexttrain+1]+' Trains at ' +strain1[nexttrain+1]
```

8/15/23 no_trains7-5-3.py makes the time display large.



New Hope Valley Railway
Tuesday, August 15, 2023

11:58 PM

**There are no trains scheduled today.
The next train rides are scheduled for
Wed 8/16: 1 Train at 10:30 AM
Sat 8/19: Operate-A-Loco NO Passengers**

[no_trains.py] no_trains7-5-3 next

To make changes to the program:

Make the changes in the Pi 3 countdown4 folder.

Copy the new file to the PC Desktop using VNC.

For example no_trains7-5-3.py

Reboot the Pi Zero and stop the autoexec.sh using Ctrl-c.

Up load the new file using VNC. It automatically goes to the Desktop of the Pi Zero. Copy the file to the countdown4 folder

cd Desktop

cp no_trains7-5-3.py /home/pi/countdown4/no_trains7-5-3.py

Now copy the filename.py file to the main file cd

countdown4

cp no_trains7-5-3.py no_trains.py

Reboot the Pi Zero.