Digital Photo Frame for the Elderly

by Andrew White

Overview

A simple to use internet connected digital photo frame that downloads and displays new images sent to its email address or GDrive account. The Digital photo frame consists of a 10.1 inch display panel, a Raspberry Pi 3 B+ and multiple python 3 scripts.

Introduction

Elderly relatives who have grand children and great grand children may not be able to see them as often as they would like, especially if they live abroad. The internet connected digital photo frame gives the elderly relatives an opportunity to see what their grand children and great grand children have been up-to when family members send photos to the photo frame.

The Digital photo frame will require no operation from the elderly relative making life easy for them, it turns on and off by itself and will automatically display its stored photos. Relatives will be able to send new images to be displayed on the photo frame by sending them to a specific email address or saving them in a specific GDrive folder.

The Digital photo frame has the following features:

- Displays the photos in a different order each day
- Changes displayed image every 10 minutes
- Displays images downloaded in the last 48 hours first in the order of photos
- Displays a randomised order of the 30 most recently downloaded images next
- Displays a random selection of any older photos last
- Checks for images in GDrive folder or email inbox every hour and downloads any new ones
- When new images have been received generate a new photo order and display the new images straight away
- Sends an email report to an administrator every week with details of photos received
- Automatically turns on in the morning and off at night

Further features of the digital photo frame:

- The system uses only the minimum required internet data meaning that it can be used with a mobile broadband device
- It only downloads and saves compatible image files
- The system will only download emails from relatives by looking for a specific subject in the email
- The photo frame software will remain stable if the internet connection drops
- The temperature of the system is logged to ensure it doesn't get too hot
- Images are scaled and rotated to display correctly

Hardware

The digital photo frame uses the following hardware:

- 10.1 inch 1200 x 1980 HDMI IPS LCD Display (https://coolcomponents.co.uk/products/10-1-inch-1200x1980-hdmi-ips-lcd-display)
- Raspberry Pi 3 B+ Model
- Anker USB Charger 4.8 A / 24 W 2 port
- Anker USB power cables x 3
- HDMI Cable
- Timer Plug
- Huawei E5330 Router with Data SIM (If no WiFi available)

Power Requirements

The IPS LCD has a working current draw of 600 mA and the Raspberry Pi a working current draw of approximately 1A when not used in intensive tasks. Therefore the 2.4 Amps provided through a single USB port of the Anker USB charger is enough for the digital photo frame application.

Photos of the Hardware

Image of the front of the digital photo frame displaying an photo.

The timer plug and the Anker USB charger. The timer plug is set to turn the power to the digital photo frame on at 8 am and off at 11:20 pm.

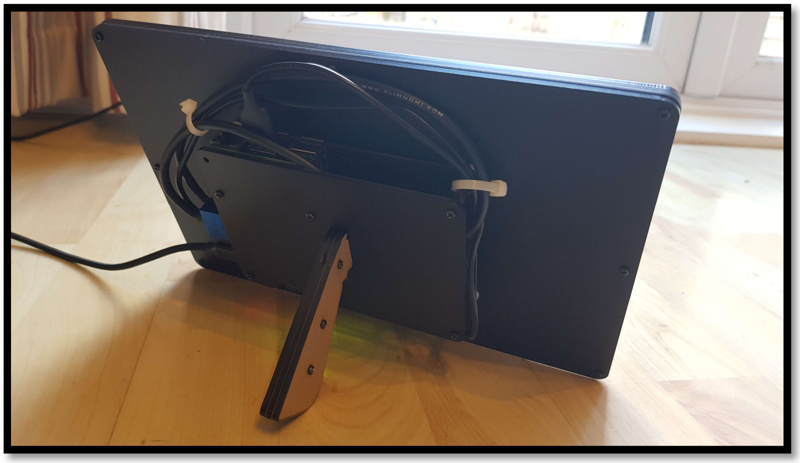
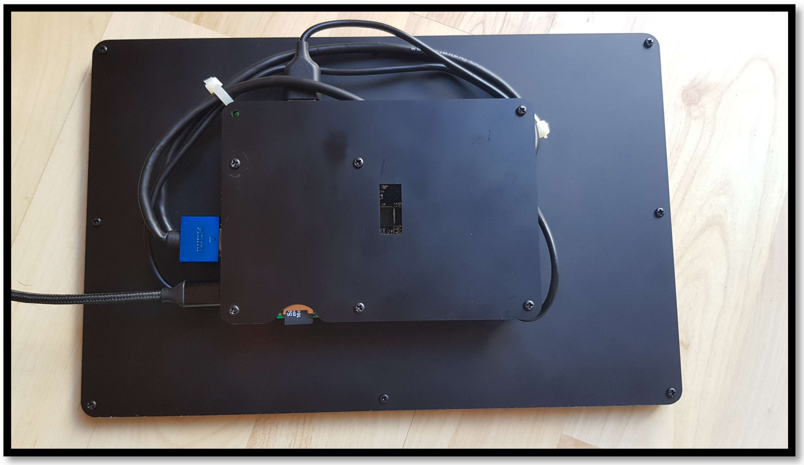


Image of the back of the digital photo frame showing how the frame will normally stand.



This image shows the back of the digital photo frame. The main power comes in from the left of the photo into the Raspberry Pi. One of the Raspberry Pi USB ports seen at the top of the photo is used to provide power to the LCD Panel. The Raspberry Pi is connected via HDMI to the LCD Panel. Both the Raspberry Pi and the driver for the LCD Panel are underneath the black bracket seen here.

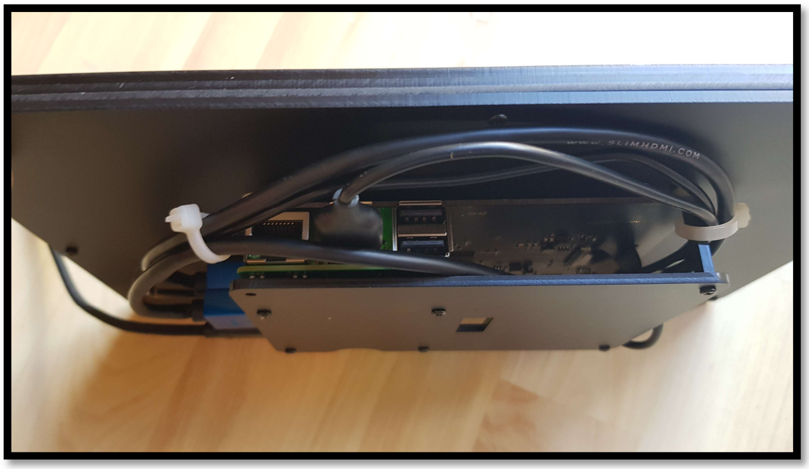


Image showing the top of the Raspberry pi and a USB cable that connects to the LCD Panel driver board to power the LCD Display.

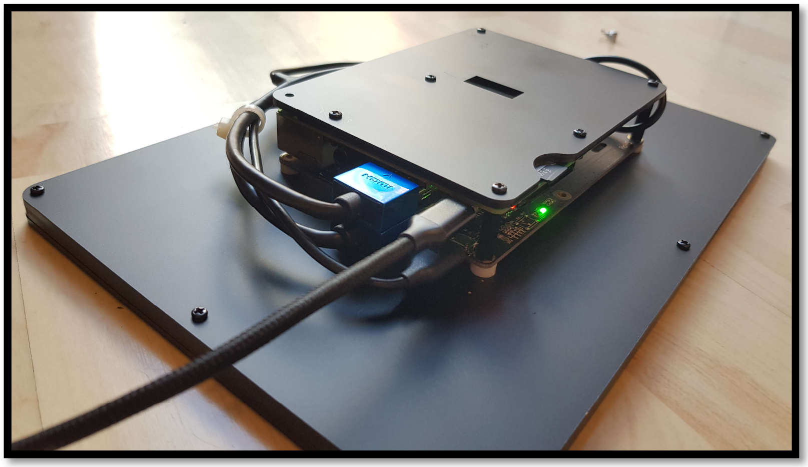


Image showing how the Raspberry Pi and LCD Panel are connected with a HDMI cable.

Software

The software runs on a Raspberry Pi 3 B+ device, running the Raspbain OS. The Raspbian OS is configured to start up the python scripts automatically at startup using bash commands written in the /etc/xdg/lxsession/LXDE-pi/autostart file. The mouse pointer and screen saver are disable using commands written in the following configuration file /etc/lightdm/lightdm.conf. The system automatically shuts down at 11:15 PM in software before the hardware timer turns off the power to the digital photo frame at 11:20 PM.

Below is a functional description of the key parts of the digital photo frame software the back end and the front end. Both use Python 3 scripts to achieve the described functionality.

Back End

main.py runs the back end process's of the digital photo frame. It performs the following operations:

- Periodically check for new photos
    - In the inbox of the GMail account
    - In the GDrive location
- Send an email report every 7 days. This report contains information on any photos that have been downloaded

Here is a more detailed run through of how the program works:

The main.py code first sets up the required settings and initialises instances of class' used in the code.

The program then enters an infinite loop. It constantly checks to see if its time to check for new images. New images are checked for every hour.

If the time is right to check for new images the program will then connect to the specified GDrive account, check the GDrive images against a list of images that have already been downloaded from GDrive. If there is an image that hasn't previously been downloaded, it will be downloaded and given a unique name. The list of images that have been downloaded from GDrive is then updated.

The program will then connect to a GMail inbox and check if there are any unread emails that have images attached and have the correct subject. If there are, they are downloaded and given a unique name and stored in the photos folder.

Both the GDrive and GMail routines will only download images if they are in JPG format, to ensure all photos are compatible with the photo frame front end.

If there is an error in connecting to GDrive or GMail, or an issue in downloading the images, such as the internet connection dropping out then the program will log this event and continue running.

When new images are downloaded with either the GDrive or the GMail services the details of that photo, such as who sent it and when it was sent are saved in the email report text file, that is sent out weekly.

The photo frame sends an email every 7 days with a text file attachment detailing what photos have been downloaded to the photo frame. At the same time each day the program will check to see if an email report should be sent, if it has been 7 days since the last report was sent then an email is sent with the report attached.

Front End

PhotoFrame2.py is the front end of the program and uses tkinter to display the images saved in the photos folder. The program performs the following operations:

- Creates a photo list for the order of images to be displayed
- Displays each image for a duration of 10 minutes before displaying the next image in the photo order list
- Checks for new downloaded images and if new images exists then creates a new photo order list with the new images at the beginning, Then displays the images in this new order

Here is a more detailed run through of how the program works:

When the program starts up it creates a pseudo random photo order list for the photos to be displayed in. To do this the code orders the files in the photos directory into the date order that they were downloaded. If any photos have been downloaded in the last 48 hours they will be put at the top of the photo order list so that the user can see the new images first.

The next 30 photos, in the photo order list will be the randomised 30 most recently downloaded images (excluding images downloaded in the last 48 hours). To complete the photo order list, out of the remaining photos in the photos directory up to 30 random images will be selected then randomised and added to the end of the photo order list.

This system is used so that the viewer will be able to see more recent photos first and more frequently. If there are not many photos in the directory to begin with, all the photos will just be randomised to create the photo order list.

Once the photo order list is created the images will start to be displayed in the order of the list at 10 minute intervals. When the next image is displayed the program checks to see if any new images have been downloaded, by the backend program (main.py). If they have, then the program will generate a new photo order list, in the same way that has been explained above, with the new downloaded images at the start of the photo order list.

When the images are loaded to be displayed the program uses the image files Exif tags that contain properties of the image file to determine the orientation of the camera when the image was taken and to rotate the image accordingly. The size of the image is then determined and the image scaled correctly so that it fits in the photo frames display.