PasswordPump II

User's Guide

Initial Setup

When you first plug the PasswordPump into a USB port you should see the following:

```
PasswordPump
Jan 12 2020
(c)2019 Dan Murphy
```

...and then

```
        Main
Master Password
```

At this point you want to decide on a master password. A master password should be something that you can enter reasonably quickly using the rotary encoder, so if you're going to use a word think of one that's made up of characters from the beginning of the alphabet. For example; *cabbages* or *Abacus*. There are many other examples. You want a word or a combination of words and numbers that are not tedious to enter via the encoder. So I typically select a word that I can enter quickly followed by a four digit number. Of course you can enter anything you like, as long as it doesn't exceed 16 characters. It is possible to change the master password once you've entered it. If you want to change it and you haven't entered any credentials that you don't want to re-enter, then simply choose Factory Reset from the main menu. If you want to preserve all entered credentials, navigate to *Settings* and then to *Change Master Psswrd*.

Ok, so if you've thought of a master password you want to use. To start the process <ShortClick> the rotary encoder (press it down without holding it down for more than 0.5 seconds). Then use the rotary encoder to scroll to the first letter you want and <ShortClick> again. Continue entering characters in this fashion until they are all entered, then <LongClick> (hold down the rotary encoder for more than 0.5 seconds), and you'll see the following:

```
        Main
Find Favorite
0 accounts
```

Remember your master password. If you forget it you'll lose access to all of the data you've entered, and short of breaking AES-256 somehow, you're not getting them back (unless you have exercised the feature that allows you to export all of your credentials to a file, which you should also encrypt).

You may also notice that when you first plug in the PasswordPump that your operating system looks for drivers to install for the device. On Windows 7 and 10 I've had mixed experiences. Sometimes everything works fine without having to take additional steps (typically with Windows 10), and sometimes I have to manually install drivers supplied by AdaFruit (typically with Windows 7). I've included those drivers in the repository for the PasswordPump, here: https://github.com/seawarrior181/PasswordPump_II. Download and run adafruit_drivers_2.4.0.0.exe if necessary. Install the drivers that are selected by default. If you install the drivers (e.g. for Windows 7), in the Device Manager you'll see *ItsyBitsy M4* in the description for the device (under Ports (COM & LPT). If you don't install the drivers (e.g. with Windows 10) you'll see *USB Serial Device* in the description for the device.

Now you're ready to start entering credentials. The easiest and best way to do that is via PasswordPumpGUI (a.k.a. PassPumpGUI_v0_7.py). It's a python program that serves up a user interface that you use to maintain credentials (account names, user names, passwords, previous password, and URL). To do that you need to use the rotary encoder to scroll down to *Edit with Computer* and <ShortClick>. Then you need to run PasswordPumpGUI.py on your computer and via it, open the correct port and start adding new sets of credentials! See the instructions included herein for setting up PasswordPumpGUI.

Navigating through the menu on the PasswordPump is explained on 5volts.org in the article for the PasswordPump (version 1.0). These instructions will eventually be updated to cover the PasswordPump II, but at this time they are the best reference for how to navigate the menus. Note that the menus for the PasswordPump v1.0 and the PasswordPump II are not the same. There are a lot more features available in the PasswordPump II, and there is also plenty of room for expansion in the future.

Troubleshooting the PasswordPump

1) *When I run PasswordPumpGUI.py, select the correct port and click Open, I see the following error message:*

```
C:\Users\someusername\Desktop>c:\python3\python
c:\PathToPasswordPumpGUI\PasswordPumpGUI.py
COM69: Adafruit ItsyBitsy M4 Express Arduino (802B:00) (COM69)
Connecting to arduino on COM69... done.
None
Exception encountered reading return value from pyReadHead; 'NoneType' object is
 not subscriptable
None
TypeError encountered in clickedOpen(); 'NoneType' object is not subscriptable
Opened port
```

This happens when you neglect to enter 'Edit with Computer' mode on the PasswordPumpII device before opening the port. If you have entered 'Edit with Computer' mode on the device and you're still seeing this error message, try power cycling the PasswordPump, restarting the PasswordPumpGUI.py program, and trying again. In the extreme situation it's necessary to restart the computer to fix issues with the port in order to resolve this problem.

Another cause of this problem is neglecting to make the required edits to the PyCmdMessenger.py file after installing it with pip or pip3. See Setting Up PasswordPumpGUI and pay special attention to the edits that are required in PyCmdMessenger.py.

2) *When I select File->Import from PasswordPump from PasswordPumpGUI.py, navigate to and select a file, some sets of credentials import but eventually I see the following error message and the GUI freezes:*

```
Error encountered reading file in ImportFilePasswordPump; 'NoneType' object is not subscriptable
```

Click on the Exit button. If the UI remains frozen, click on the [x] close icon in the top right hand side of the window (under Windows). On the PasswordPump, long click the rotary encoder. If that doesn't return control, click on the reset button on the PasswordPump. Login to the device and re-select 'Edit with Computer'. Make sure that, in the export file, none fields contain a | (pipe) or a ~ (tilde). Make sure that none of the URLs end in / (forward slash). Launch PasswordPumpGUI.py and, once you select and open the correct port, re-try the import operation.

3) *When I change an existing account name a new account is created with the new account name and no attributes populated, and the old account remains.*

There's presently no easy way to change the account name. The best way to rename an account is to insert a new account with the desired account name and attributes, and then delete the old account. If you edit the account name of an existing set of credentials, a new account is created with the new account name when focus leaves the account name field, and all of the other attributes are initialized to be empty. The previous account remains (you may delete it if you like). I've not decided yet if this is the desired long term behavior.

4) *When I enter an account name, username, password, or any other field that contains a ~ (tilde), a | (pipe), a " (double quote), or a , (comma) that character is changed to a # (hashtag). This behavior is also observed when importing data with tildes, pipes, double quotes and/or commas.*

Tildes, pipes, double quotes and commas are not supported in any of the fields. You'll need to eliminate them from your account name, username, password or other fields. If you enter them from the PasswordPumpGUI they are automatically changed to # (hashtag). It's not possible to add them via the rotary encoder.

5) *I'm using a Mac, Linux, or some other operating system other than Windows 7, Windows 10, or Ubuntu, and things don't seem to be working correctly.*

   Congratulations, you are the first person to use the product on your platform. Kindly let the PasswordPump developer (dan-murphy@comcast.net) know about your issue(s) so that we can work on a solution. Your help is greatly appreciated. Alternatively use the forums at 5volts.org to solicit help.

6) *When navigating between accounts via the PasswordPumpGUI program the fields get out of synch; for example the account name appears on the Username text box, or fields are otherwise out of sync.*

   The most likely cause of this problem is a | (pipe) character embedded in the account name, username, password, old password, or URL fields. To find the offending field edit the account via the rotary encoder on the PasswordPump device. On the PasswordPumpGUI you can navigate between accounts, and when the alignment of fields looks wrong, take note of the previous account visited. The problem is most likely in one of the fields of the previously visited account. You can open up Notepad (if you're running Windows), and paste the username, password, account name, URL and old password into Notepad (using the PasswordPump device). Note if any of the fields have an embedded | (pipe) character.

7) *After clicking on or navigating to an account via PasswordPumpGUI, the following error message is displayed in the python console:*

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python3\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\repos\murphyrepo\dev\python\PassPumpGUI\PassPumpGUI_v0_7.py", line 50
9, in clickedNext
    OnEntryDownNoEvent()
  File "c:\repos\murphyrepo\dev\python\PassPumpGUI\PassPumpGUI_v0_7.py", line 56
8, in OnEntryDownNoEvent
    OnEntryDown(0)
  File "c:\repos\murphyrepo\dev\python\PassPumpGUI\PassPumpGUI_v0_7.py", line 57
8, in OnEntryDown
    clickedLoad()                                                    # cal
ls getRecord()
  File "c:\repos\murphyrepo\dev\python\PassPumpGUI\PassPumpGUI_v0_7.py", line 63
4, in clickedLoad
    getRecord()
```

```
  File "c:\repos\murphyrepo\dev\python\PassPumpGUI\PassPumpGUI_v0_7.py", line 70
3, in getRecord
    response = c.receive()
  File "c:\repos\murphyrepo\dev\python\PassPumpGUI\PyCmdMessenger\PyCmdMessenger
.py", line 280, in receive
    raise ValueError(err)
ValueError: Number of argument formats must match the number of recieved argumen
Ts.
```

This happens when there are corrupt values in the old password field. To fix it, simply set focus in the Old Password text box and set focus on the account or password text box to save an empty string to old password on the PasswordPump. Do not hit tab after clicking in the Old Password text box or you might corrupt the value in the URL field. I believe the defect that created this situation is addressed so if you encounter this problem please report it to me.

8) *I entered a duplicate account name and lost all of my credentials for all accounts.*

While not possible solely via the PasswordPumpGUI, it is possible to enter a duplicate account name via the rotary encoder on the PasswordPump or via a combination of the PasswordPump and the PasswordPumpGUI. When you delete one or both of these accounts the PasswordPump can become corrupt, so it's important to have a backup of all of your sets of credentials so that you can restore back to a known good state. This is an open defect that I am working to address.

9) During import of a large PasswordPump format file, the process stops with an error.

There is a defect in the PasswordPumpGUI whereby an error is occasionally encountered during the import of a (typically) large PasswordPump formatted file. This problem is intermittent, and therefore difficult to pin down. For now the best approach to dealing with it is to just start from the beginning; i.e. factory reset your device and re-initiate the import operation. In terms of frequency, I estimate that for every account you import there is a 1 in 300 chance of encountering the error. If you encounter this problem with greater frequency please contact me.

10) When I press on the screen hard enough the PasswordPump resets itself.

The reset button on the ItsyBitsy M4 is located under the screen, so if you press on the screen hard enough you'll actuate the reset button. Don't do that. To reset the device use the button on the bottom of the device instead.

11) A certain field of a certain account will not, under any circumstances, store a particular value for that field. For example, I am trying to set my Passoword to abcdefg. I set focus to the Password field, enter abcdefg. When I return to that account the Password field is blank. If I put any other value in the field; e.g. abcdefgh, or abcdef, this freaky

behavior doesn't happen. This is annoying, what's going on?

This happens under certain circumstances and is related to how we encrypt and decrypt passwords (and all other fields in the account, for that matter, with the exception of style and group). The solution to this problem is tedious; you should either change the value that your trying to store in that field, or you should delete the account and re-insert it. This problem is more of an annoyance when you're importing a large number of credential sets, because there's no way to know if a certain field on a certain account was dropped. Fortunately it doesn't happen very often.

12) *On one of my account names, the saved account name is shorter than that which I entered, and I've entered less than 31 characters.*

Account name can only be up to 31 characters long. However, sometimes they are truncated even further. This is a cousin to the problem above. It doesn't happen very often, but it can happen. The workaround is to either accept the shortened name, or to change the account name altogether. Remember that the account name isn't the username, it's not typically supplied when you're authenticating, so you can make it whatever you want. The username, however, is more of a issue, but this problem does not affect the username field.

13) *When my account name has commas in it, if I visit the account name field in the PasswordPumpGUI, after I reload the accounts (exit and restart PasswordPumpGUI), the commas are replaced with hashtags and all of the other fields are blank.*

Don't import credentials with commas. If you have an account with a comma do not set focus on the account field in the PasswordPumpGUI. A fix is underway.

Features

(bolded items are new PasswordPump II features)

- Store up to 254 sets of credentials
- Authenticate with a 16 character master password
- Search for accounts
- Data entry via rotary encoder or keyboard and serial monitor, or via client **Python GUI** running in Windows, Ubuntu, or MacOS.
- Send username and password as if typed in keyboard. Can also send URL, old password and account name.
- Add account name, username, password (generated or not), **URL**, **old password**
- Accounts added in alphabetical order
- Edit existing username, password, URL, style (inter-username/password character, <Return> or <Tab>), **old password**

- **Automatically saves old password if it's not already populated**
- Delete account
- Generate 31 character random password from the PasswordPump **of via the client GUI**.
- Backup all accounts to a second encrypted external EEprom
- Logout / de-authenticate via menu
- Factory reset via menu (when authenticated) wipes out all credentials
- Configurable password display on or off
- **Configurable** failed login count factory reset (3, 5, 10 or 25)
- **Configurable** automatic logout after count of minutes (30, 60, 90, 120, 240, 1 or Never)
- **Configurable** RGB LED intensity (high, medium, low or off)
- All passwords (except master password) are encrypted w/ **AES-256**; master password is hashed w/ SHA-256.
- All encrypted accounts and the hashed master password are salted
- The device is not vulnerable to standard password attacks. See disclosures
- **Change master password**
- **Export to PasswordPump formatted CSV file**
- **Import from PasswordPump formatted CSV file**
- **Import credentials from Chrome export**
- **Import credentials from KeePass export**
- **Associate credentials with groups for better organization; search by group (Favorites, Work, Personal, Home, School, Financial, Mail or Custom)**
- Decoy password feature that automatically factory resets the device if entered (e.g. while the user is under duress)
- **Pre-auto-logout indicator/countdown via red and blue flashing RGB LED.**

Disclaimers

- The PasswordPump is not secure from keylogging attacks (https://en.wikipedia.org/wiki/Keystroke_logging). Keylogging attacks are capable of stealing passwords that are entered through your keyboard. All data sent to your computer with the PasswordPump enters the computer as if through the keyboard. Therefore you should remain diligent about protecting yourself from these kinds of attacks. See the countermeasures section of the Wikipedia link provided above.
- The contents of the EEprom chips on the PasswordPump are encrypted with AES-256, and the master password is hashed with SHA-256. The unhashed master password serves as the encryption key (along with 16 bytes of salt). The credentials are also salted. Nevertheless, if somebody with nefarious purposes obtains access to your PasswordPump it's best to assume that all of your credentials have been compromised. It is possible to move the encrypted contents of the EEprom to an operating system file

by using a USB programmer (e.g. TL866II Plus). This would allow an attacker to circumvent the protections built into the device that prevent more than 3, 5, 10 or 25 failed login attempts before the credentials are wiped. I will consider removing this advice when the device's software has been subjected to a rigorous code review by encryption industry experts.

- Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the creator of this device and software be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of the use of the PasswordPump including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, personal injury, death or any and all other damages or losses.

Menu Navigation on the PasswordPump

You move through the menu items by turning the rotary encoder, clockwise to move down the list and counter clockwise to move up. Account names are stored in alphabetical order. To select an item you click down on the rotary encoder (short click). To backup you hold the rotary encoder down for more than a half second (long click).

Master Password
Find Favorite
Find All Accounts
  [scroll through accounts list]
    Send Password <RET>
    Send User & Pass
    Send URL
    Send User Name
    Send Pass (no <RET>)
    Send Account
    Edit Credentials
      Edit Account Name
      Edit User Name
      Edit Password
      Edit URL
      Indicate Style
      Assign Groups
        Favorites
        Work
        Personal
        Home

- School
- Financial
- Mail
- Custom
- GeneratePassword
- Save to Old Password
- Delete Credentials [confirm]
- Send Old Password

Find By Group
- Favorites
  - [same as under Find All Accounts]
- Work
  - [same as under Find All Accounts]
- Personal
  - [same as under Find All Accounts]
- Home
  - [same as under Find All Accounts]
- School
  - [same as under Find All Accounts]
- Financial
  - [same as under Find All Accounts]
- Mail
  - [same as under Find All Accounts]
- Custom
  - [same as under Find All Accounts]

Add Account
- Account Name
- Edit User Name
- Edit Password
- Indicate Style
- GeneratePasswrd

Logout

Backup/Restore/Imprt
- Backup EEprom [confirm]
- Restore EEprm Backup [confirm]

Settings
- Keyboard ON/OFF
- Show Passwrd ON/OFF
- Decoy Password ON/OFF
- RGB LED Intensity
  - High
  - Medium
  - Low

Off
Timeout Minutes
30
60
90
120
240
Never
1
Login Attempts
3
5
10
25
Change Master Password
Factory Reset [confirm]

Operation of the PasswordPump via Rotary Encoder

(these are the operating instructions for the PasswordPump v1.0, they will eventually be updated for the PasswordPump II)

To turn the device on you simply plug it into a USB port/receptacle using a micro USB Micro-B plug to USB-A plug cable, the same cable that you'd use to charge an Android phone. The first time you plug it in a driver might need to be installed. The driver is available for download in the source code repository here: https://github.com/seawarrior181/PasswordPump_II. If I shipped the device to you it arrives already flashed with the program.

The first time you power the device on you'll see :

> *Master Password*
> *March 11 2019 (or the date of the most recent compile)*

At this point you'll want to enter your master password. Try to select a password that can be more quickly entered into the device. It should be a combination of upper and lower case, with numbers and maybe a symbol or two. I like to pick a password that can be typed almost entirely with my left hand, I find they are easier to input via the rotary encoder. You should select a strong password; a combination of letters, upper and lower case, numbers, and special characters, between 7 and 31 characters long. To enter a character turn the rotary encoder until the character appears and then press the rotary encoder down (short click) to select the

character. There's presently no way to back up if you make a mistake so be careful. Once the entire master password has been entered long click the device (click down the rotary encoder for more than 1/2 a second). You've just entered the master password and now you're ready to enter a set of credentials. Don't forget your master password, it's the only way to recover your encrypted credentials short of cracking SHA-256 or AES-256.

You move through the menu items by turning the rotary encoder, clockwise to move down the list and counter clockwise to move up. Account names are stored in alphabetical order. To select an item you click down on the rotary encoder (short click). To backup you hold the rotary encoder down for more than a half second (long click).

Note: The following instructions describe the easiest way to enter credentials if you don't have access to the PasswordPumpGUI or if it's not working correctly. The easiest way to enter credentials is via the PasswordPumpGUI, and it's fairly self-explanatory, so use that method if possible.

Adding Credentials via Keyboard

You can add credentials via the PasswordPump by entering them directly with the rotary encoder or by using a keyboard in combination with a serial terminal. To add a set of credentials via the keyboard you need to open a serial terminal. The one that works best for me is the Arduino serial terminal. So if you open the Arduino IDE go to Tools->Ports and select the *Arduino/Genuino Micro* port. Then select Tools->Serial Monitor (or Ctl+Shift+M). Next, on your PasswordPump navigate down to Keyboard OFF and change it to Keyboard ON with a short click. Navigate back up to Add Account and short click. You'll see:

Account Name
Add Account

Short click, then switch back to the Arduino Serial Terminal and enter the account name, followed by the return key. Then long click on the Password Pump. You should now see:

Edit Username
[the account name you entered]

Short click again, switch back to the Arduino Serial Terminal and enter the username, followed by the return key. Then long click on the Password Pump. You should now see:

Edit Password
[the account name you entered]

Short click again, switch back to the Arduino Serial Terminal and enter the password, followed by the return key. Then long click on the Password Pump. You should now see:

Indicate Style
[the account name you entered]

Short click again and use the rotary encoder to specify either 0 or 1. Specify 0 if, while supplying username and password, the Password Pump should send a carriage return after sending the username and before sending the password. Specify 1 if, while supplying username and password, the Password Pump should send a tab after sending the username and before sending the password. Then long click on the Password Pump. You should now see:

Account Name
[the account name you entered]

Long click again and you'll see:

Find Account
[the account name you entered]

You've finished entering the credentials.

Note that you can also enter credentials using just the rotary encoder. Keyboard can be ON or OFF, it doesn't matter. Simply enter the credentials using the rotary encoder in a fashion similar to how you entered the master password.

Sending Credentials

Navigate to Find Account and short click. Use the rotary encoder to scroll through the list of credentials you've entered. When you've found the account name associated with the credentials you want to send to your computer, place the input focus in the username text box in the window prompting you for credentials on your computer. On the Password Pump you should see:

Send User & Pass
[the account name you selected]

Short click to send the username, a carriage return or a tab character (depending on the style setting), and then the password. If you selected the correct style you should now be logged in to your account / application.

If you only want to send the password to the computer, followed by a carriage return, scroll down once using the rotary encoder until you see:

Send Password <RET>
[the account name you entered]

And short click to send the password and the carriage return character.

Similarly you can send just the username or just the account name.

Editing Credentials

To edit a set of existing credentials first decide if you're going to edit the credentials via the keyboard or just the rotary encoder. If you're going to edit the credentials via the keyboard follow the instructions in *Toggling Keyboard Entry.* Then use *Find Account* to navigate to the account you want to edit and short click. Then scroll down to *Edit Creds* and short click. Then scroll to the attribute you want to edit; *Account Name, Edit Username, Edit Password*, or *Indicate Style*. Now short click. Use the keyboard to re-enter the attribute in the fashion described in *Adding Credentials*, or just use the rotary encoder to re-enter the attribute. Then long click to save the change. If you are generating a new password for the account then follow the instructions in *Generating a Password.*

Deleting Credentials

Make sure you have a current EEprom backup. Navigate to *Find Account* and short click. Use the rotary encoder to select the account that you want to delete, and short click. Using the rotary encoder scroll down to *Delete Acct* and short click. Confirm your desire to delete the account by selecting *Y* with the rotary encoder and short clicking. The account is gone now and it's wiped from the primary EEprom chip. It isn't wiped from the backup EEprom yet, so if you accidentally delete an account, and you have a recent backup, you can restore the backup and the account will reappear. Navigate to *Find Account* and verify that your account is deleted. If you're not able to scroll through all of your accounts, a intermittently occurring defect has occurred and the linked list that manages the display of all of the accounts is corrupted. Restore the latest backup from EEprom. If you backup the EEprom immediately after deleting the account it is also wiped from the secondary EEprom.

Generating a Password

Read through all of these instructions before attempting to change your password to a new generated password. The most powerful feature of the PasswordPump is it's ability to generate random 31 character passwords and remember them. These passwords are extremely difficult to guess and are not as vulnerable to brute force attempts to break into an account. Before performing this operation you should be sure that you have a current backup of all your credentials. To generate a password for an account simply find the account via *Find Account* and select the credentials by short clicking on the account name. In your application on your

computer navigate to the change password feature and place input focus in the Old Password text box. On the PasswordPump navigate to *Send Password* (NOT *Send Password <RET>*) and short click. In your application on your computer place input focus in the new password text box by hitting the <TAB> key. In the PasswordPump scroll down to *Edit Creds* and short click, then scroll down to *Gen Password* and short click. This changes the password to a randomly generated series of 31 characters. Now long click once, navigate to *Send Password* (NOT *Send Password <RET>*) and short click. If you need to confirm the new password then place input focus on that text box in the application on your computer and short click again. Confirm your password change by hitting the return key or otherwise clicking on the appropriate button. You now have a random 31 character password on the account, and the only place where that password exists is on the encrypted EEprom chip on your PasswordPump. At this point it's a good idea to *Backup to EEprom* and *Backup to a File,* and to be sure that you can somehow recover from a lost password on that account. Warning: If the attempt to change your password fails because the old password is not accepted be aware that you have just overwritten the old password with your new generated password. To restore the old password you'll need to either *Restore a Backup from EEprom* and try again, or go to the encrypted backup file on your thumb drive to get the current password for the account, or recover the password from the account using whatever mechanism is available to you via the application or web site. Think ahead and be careful so that you don't lock yourself out of your account!

Logging Out

When you want to log out of the device navigate to *Logout* using the rotary encoder and short click. The RGB led changes from green to blue. You're now logged out and must enter the master password again in order to use the device. If you're walking away from your computer to get coffee it's good practice to lock your computer and logout of the PasswordPump.

Toggling Keyboard Entry

Navigate to *Keyboard ON/OFF*. Short click to toggle the setting. When the keyboard is on you may enter credentials via the keyboard using the process described in *Adding Credentials.* Keep the keyboard set to OFF when you're not entering credentials. This setting is not saved when the device powers off and the default is *Keyboard OFF*.

Showing/Hiding Passwords

Using the rotary encoder navigate to *Show Psswrd ON/OFF.* Short click to toggle the setting. This setting is saved when you log out and power down the device.

Backing Up to EEprom

On the Password Pump navigate to *Backup EEprom* using the rotary encoder. Short click, then confirm that you want to backup from the primary EEprom to the secondary EEprom by selecting *Y* with the rotary encoder and short clicking. The RGB will be yellow while the backup is taking place, and then change back to green.

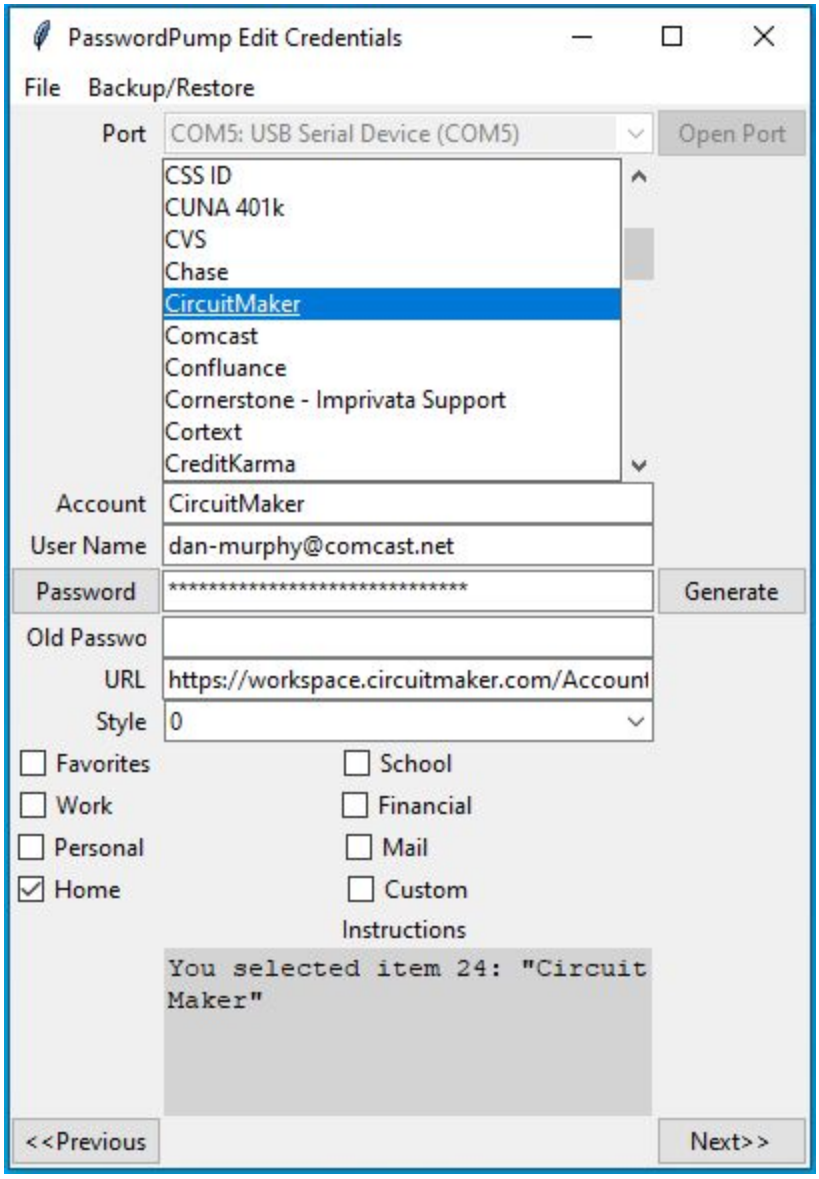
Restore a Backup from EEprom

If you decide that you want to restore the EEprom backup (or, in other words, have the contents of the secondary, backup EEprom overwrite the contents of the primary EEprom), then navigate to *Restore Backup* on the PasswordPump. Short click and confirm the operation by selecting *Y* with the rotary encoder and short clicking. The RGB led will turn yellow until the operation is complete, then it changes back to green. The master password remains the same.

Performing a Factory Reset

You want to wipe out all of the encrypted credentials on the primary and backup EEprom and factory reset the device. On the PasswordPump navigate all the way down to *Reset* using the rotary encoder. Short click. Confirm that you want to factory reset the device and clear all of the credentials and the master password from both EEprom chips by selecting *Y* with the rotary encoder and short clicking. The RGB will flash blue and red slow and then fast while the device is factory resetting, then change to blue. At this point you can enter a new master password. Note that a Factory Reset also wipes out the credentials stored on the backup EEprom.

Setting Up PasswordPumpGUI



Download Python 3.8 for your computer's operating system from here: https://www.python.org/downloads/release/python-381/. After installing Python 3.8, use pip to install the *tendo* and *PyCmdMessenger* packages:

```
pip install tendo
pip install PyCmdMessenger
```

Edit PyCmdMessenger.py (C:\Python3\Lib\site-packages\PyCmdMessenger-0.2.4-py3.8.egg\PyCmdMessenger\PyCmdMessenger.py [or similar] on Windows, /home/*username*/.local.lib/python3.5/site-packages/PyCmdMessenger/PyCmdMessenger.py [or similar] on Ubuntu);

Starting on line 25:

```
field_separator="~"
command_separator="|"
escape_separator="\\"
```

On line 361 and 362 of the same PyCmdMessenger file add the following (only what's highlighted/bolded):

```
# Range check
if value > 255 or value < 0:
      err = "Value {} exceeds the size of the board's
byte.".format(value)
      raise OverflowError(err)
if value == 92:        # this is line number 361
      value = 1
return struct.pack("B",value)
```

Be careful about the indentation, python is very picky. Use spaces to indent, not <tab>. Otherwise you'll see the error **`inconsistent use of tabs and spaces in indentation`** when you try to run the program.

Now you can download PassPumpGUI_v0_7.py from this location: https://github.com/seawarrior181/PasswordPump_II/blob/master/PassPumpGUI_v0_7.py. Save the file to your desktop. Then create PasswordPumpGUI.bat and save that to your desktop as well. Here are it's contents (assuming you're on Windows and you installed Python 3.8 to C:\python3)::

```
c:\python3\python c:\yourUsername\desktop\PassPumpGUI_v0_7.py
```

Substitute *c:\python3* from above with the location where you installed Python 3.8, and substitute *yourUsername* with your username. Now place your PasswordPump into *Edit with Computer* mode and you should be able to double click on PasswordPumpGUI.bat from your desktop to launch the PasswordPumpGUI python program. Open the correct port and you'll be able to edit credentials from the GUI.

Tips & Tricks

- Do not make a habit out of unplugging the device from it's micro-B USB port. Instead unplug the end of the cord that plugs directly into the computer (USB A), and leave the device plugged into the cord. This reduces the wear and tear on the device's micro USB port and will extend the life of the unit. I have seen similar micro-B USB ports fail.
- After you create a KeePass or a Chrome export file, and before importing into the PasswordPump, edit the .csv file and make sure that none of the accounts have embedded commas (,), pipes (|), tildes (~), or slashes(/\). These characters tend to create problems and I am working on solutions. After removing the problematic characters save the .csv file before importing.
- If you have many accounts, associate the accounts you use the most with the Favorites group. Of these favorite accounts, name the accounts you use the most with an _ (underscore) so that they will sort to the top of the favorites list. I use this technique to identify my MS Active Directory credentials, which I have to supply in many places. That account is named _Active Directory, so it always sorts to the top. After I login to the device I can short click three times and my _Active Directory password is typed into the computer via the PasswordPump. Using this technique my most frequently used password is always a few clicks away. Even after I have sent a different password, I can quickly send the 'default' password with three long clicks followed by three short clicks. Just be sure your input focus is on the password field.
- A master password should be something that you can enter reasonably quickly using the rotary encoder, so if you're going to use a word think of one that's made up of characters from the beginning of the alphabet. For example; *cabbages* or *Abacus*. There are many other examples. You want a word or a combination of words and numbers that are not tedious to enter via the encoder. So I typically select a word that I can enter quickly followed by a four digit number. Of course you can enter anything you like, as long as it doesn't exceed 16 characters. Even a four digit pin might be secure enough for you.
- Remember to *Backup to EEprom* after changing attributes of existing credentials or after inserting new credentials. I usually confirm that I can navigate through all existing accounts forwards and backwards before executing a *Backup to EEprom* operation, just to be sure the linked list that contains all of the credentials isn't corrupt. I haven't seen this problem for a long time.
- Immediately after performing a *Change Master Password* operation, confirm that you can navigate through all the accounts forwards and backwards, then perform a *Backup to EEprom* operation. If for some reason your credentials look corrupted after a *Change Master Password* operation, *Restore EEprom Backup* will restore your credentials to the device with the original master password.
- Before removing the EEprom chip(s) from the device, power it off by unplugging it from your computer.
- Instead of executing a *Restore EEprom Backup* operation you can swap the positions of the EEprom chips instead.
- Maintain a third EEprom backup and secure it in a safe place. You can purchase extra 25LC512 chips on AliExpress, Amazon or Ebay. If your data are corrupted and you cannot *Restore EEprm Backup,* you can insert this backup into the primary EEprom

position (the top chip). Don't forget to perform a *Backup EEprom* operation twice; once to create a new offline backup and once to create a new online backup, for safety.

- Use the PasswordPumpGUI to create a PasswordPump formatted .csv file. Encrypt this file and/or store it on an encrypted thumb drive, and store the thumb drive in a safe or safe deposit box (perhaps alongside your EEprom backup). If your PasswordPump's data becomes corrupted you can perform a *Factory Reset* operation and then Import Password Pump file via the PasswordPumpGUI. If you are diligent about backing up your credentials in this manner (and in the manner described in the previous bullet) you can use the PasswordPump as your system of record for all of your credentials.
- EEprom chips can be moved to another PasswordPump device and will continue to function without modification. The master password moves with the EEprom chip. The hashed master password and salt are stored in the external EEprom chips.
- If the Old Password field is empty, it's automatically populated with the existing password when the Generate password button is clicked in the PasswordPumpGUI or directly via the device. If the field is *not* empty, clicking on the Generate password button will *not* move the existing password to the Old Password field, but it will overwrite the existing password with the generated password. If the Old Password field is populated and you want the existing password to move to the Old Password field after clicking the Generate button (which would be the typical use case), then blank out the Old Password and move input focus off of that field (so that the change is saved to the device) *before* clicking on Generate. The Old Password field is intended to protect the user from the situation whereby a password change is being made, the Generate button is selected to generate a new password, but the application or website for which your changing the password does not accept the newly generated password for any reason and you have therefore lost the currently active password. By proper use of the Old Password field, in this situation, you have not lost the currently active password, it is in the Old Password field, and you can still use it to continue trying to reset the password.
- Use of the PasswordPumpGUI currently requires the installation of Python 3.8. At some point in the future an .exe will be created so that this requirement can be removed.
- Source code is located here: https://github.com/seawarrior181/PasswordPump_II
- Send any issues and suggestions to dan-murphy@comcast.net.

Uploading the Latest Firmware to the PasswordPump

To burn the latest version of the firmware, first download PasswordPumpM4_01.ino.bin from https://github.com/seawarrior181/PasswordPump_II and save it to C:\temp. Find where the **bossac** utility is installed on your system and modify the path below as per its location. Identify the port to which the ItsyBitsy/PasswordPump is connected (you can use the Device Manager to do this) and substitute the correct port number in the command below. Double click on the reset button at the bottom of the PasswordPump. The RGB LED will slowly dim and brighten in blue

over and over again. Sometimes it takes 30 seconds or so for the device to go into this mode; if the RGB LED is quickly blinking blue, wait for it to switch into the mode whereby the LED is slowly fading in and out in blue. From a command window (<Alt><Esc>cmd<Return>) execute a command similar to the following:

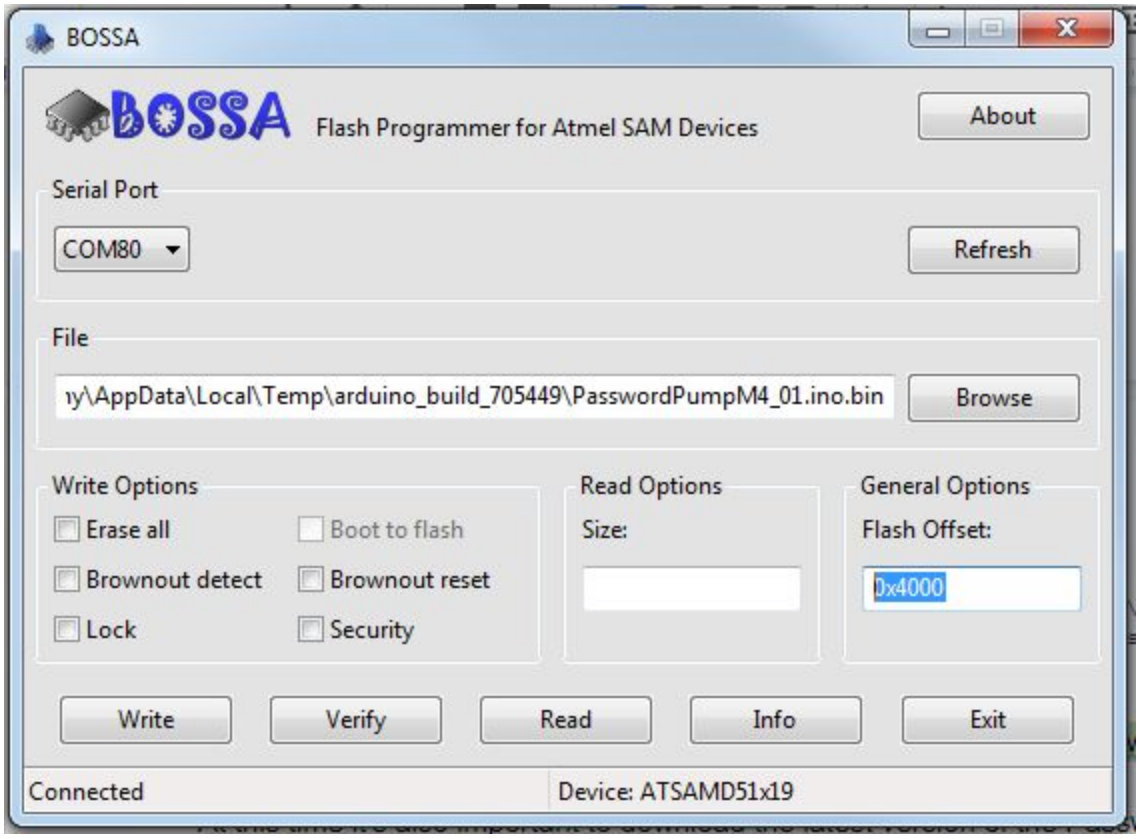
```
C:\Users\useraname\AppData\Local\Arduino15\packages\arduino\tools\bossac\1.8.0-48-gb176
eee\bossac -i -d --port=COM52 -U -i --offset=0x4000 -w -v
C:\Temp\PasswordPumpM4_01.ino.bin -R
```

You should observe output that reflects that the PasswordPump's firmware has been updated.

At this time it's also important to download the latest version of the PasswordPumpGUI; which is here: https://github.com/seawarrior181/PasswordPump_II/blob/master/PassPumpGUI_v0_7.py.

At a point in the near future the version numbers will be synchronized to make it more obvious that a particular version of the firmware only works with a particular version of the PasswordPumpGUI.

If you don't want to install the Arduino IDE but you still wish to burn the latest version of the firmware, note that it's also possible to install the BOSSA utility to accomplish this. You can download it from https://github.com/seawarrior181/PasswordPump_II and install it on your MS Windows or Apple Mac OS X computer. Again, double click on the reset button on the PasswordPump so that the blue LED slowly fades on and off before burning the firmware. After running the utility, to burn the firmware, use all of the defaults except **specify a flash offset of 0x4000**. Click *Write* to write the firmware to the device, then click *Verify* to verify that it was written correctly. Finally click the reset button once to start using the PasswordPump.



RGB Colors and Meanings

Color	Meaning
Green	Logged in
Orange	Backing up EEprom memory
Alternating Blue and Red	Initializing EEprom or auto logout pending
Purple	Sending creds, backing up to EEprom, editing with computer.
Red	Error backing up or initializing EEprom, failed login attempt(s).
Yellow	Error backing up or initializing EEprom
Blue	Not logged in

Error Codes

These error codes are observed on the PasswordPump device, typically on the third line, when something goes wrong:

000 - SSD1306 allocation failed (only visible via serial)

001 - Error navigating Off On menu
002 - Error navigating main menu
003 - Error navigating edit credentials menu
004 - Error navigating send credentials menu
005 - Error navigating settings menu
006 - Error showing credential values
007 - Unrecognized event
008 - Invalid state when showing Off On menu
009 - Invalid login attempt maximum
010 - Out of space
011 - Corruption found
012 - Out of space during import
013 - Failed to open file for import
014 - Failed to mount FAT file system during import
015 - Failed to initialize flash during import
016 - Invalid RGB LED Intensity position
017 - Invalid maximum login attempt count
018 - Invalid logout timeout value
019 - Invalid keyboard, show password or decoy password value
020 - Account name keeps encrypting to 255 in first char during import
021 - User name is too long on import
022 - Password is too long on import
023 - Web site is too long on import
024 - Account name is too long on import
025 - Invalid group specified
026 - Invalid search group specified
027 - Invalid group menu item specified
028 - Invalid state during event single click
029 - Invalid state encountered during rotate counter clockwise event
030 - Invalid state encountered during rotate clockwise event
031 - Empty credentials found in linked list
032 - Corrupt linked list
033 - Corrupt linked list in FindAccountPos.
034 - Failed to initialize flash during PasswordPump CSV file import
035 - Group length is greater than one
036 - Too many fields found in PasswordPump CVS file during import
037 - Failed to open PasswordPump CVS file for import
038 - Invalid position in file menu
039 - Encrypted account name starts with 255, fixing...
040 - Invalid position when returning to a find by group menu
041 - Corrupt link list encountered while counting accounts

Fixed Costs

1 AdaFruit ItsyBitsy (32-bit ARM®, SAMD51 Cortex®-M4F MCU)*	$14.95
2 MICROCHIP - 25LC512-I/P - 512K SPI™ Bus Serial EEPROM DIP8	3.80
1 SSD1306 I2C LED display 128x32 pixels.	2.23
1 Custom PCB	1.50
1 micro USB to USB cable 100cm	0.69
1 Rotary Encoder	0.46
1 plastic knob for rotary encoder	0.11
2 IC DIP Sockets, 8 pins each	0.10
Solder	0.10
1 RGB LED diffused 5mm	0.03
3 220ohm resistors	0.01
2 4.7kohm resistors	0.01

Total Parts	$23.99
	======
Shipping to UK from USA	$14.25
Total Parts & International Shipping	$38.24

+ Labor for assembly, packaging & the case

Assembly time, including kitting and burning firmware, 45 minutes.

*Retail price from Adafruit

Connections

These are the connections made by the custom PCB, i.e. connections that must be made if you're building the project from a breadboard.

ItsyBitsy M0/M4

Num	Name	Connect To / Notes
1	RS	reset button
2	3V	25LC256 Prim Pin 3 & 25LC256 Secondary Pin 3
3	AREF	
4	VHI	
5	A0	
6	A1	

7	A2	
8	A3	
9	A4	220 Ohm resistor->RGB LED Pin 3
10	A5	
11	SCK	25LC256 Prim Pin 6 & 25LC256 Secondary Pin 6
12	MO	25LC256 Prim Pin 5 & 25LC256 Secondary Pin 5
13	MI	25LC256 Prim Pin 2 & 25LC256 Secondary Pin 2
14	2	
15	En	
16	swdio	
17	swclk	
18	3	
19	4	
20	RX	
21	TX	
22	SDA	SSD1306 SDA, 4.7k Ohm resistor->ItsyBitsy Pin 31
23	SCL	SSD1306 SCL, 4.7k Ohm resistor->ItsyBitsy Pin 31
24	5! (VHI Out)	220 Ohm resistor->RGB LED Pin 4
25	7	Rotary Encoder Pin 3
26	9	Rotary Encoder Pin 1
27	10	25LC256 Secondary Pin 1 Chip Select
28	11	25LC256 Primary Pin 1 Chip Select
29	12	Rotary Encoder Pin 4
30	13	220 Ohm resistor->RGB LED Pin 1
31	USB	SSD1306 VCC
32	G	RGB LED Pin 2, 25LC256 Prim & Secon Pin 4, SSD1306 Pin 1, Rotary Encoder Pins 2 & 5
33	BAT	

2 25LC512 (External EEprom)
Tested Part: MICROCHIP - 25LC512-I/P - 512K SPI™ Bus Serial EEPROM DIP8

25LC512 Primary

Num	Name	ConnectTo	Note
1	CS	pin 28 ItsyBitsy	Chip Select Input
2	SO	pin 13 ItsyBitsy	MISO - Serial Data Output
3	WP	pin 2 ItsyBitsy	Write Protect
4	Vss	pin 2 ItsyBitsy	Ground
5	SI	pin 12 ItsyBitsy	MOSI - Serial Data Input
6	SCK	pin 11 ItsyBitsy	SCLK - Serial Clock Input
7	HOLD	pin 2 ItsyBitsy	Hold Input
8	Vcc	pin 2 ItsyBitsy	Supply Voltage

25LC512 Secondary

Num	Name	ConnectTo	Note
1	CS	pin 27 ItsyBitsy	Chip Select Input
2	SO	pin 13 ItsyBitsy	MISO - Serial Data Output
3	WP	pin 2 ItsyBitsy	Write Protect
4	Vss	pin 2 ItsyBitsy	Ground
5	SI	pin 12 ItsyBitsy	MOSI - Serial Data Input
6	SCK	pin 11 ItsyBitsy	SCLK - Serial Clock Input
7	HOLD	pin 2 ItsyBitsy	Hold Input
8	Vcc	pin 2 ItsyBitsy	Supply Voltage

Rotary Encoder

1 2 3
4 5

Num	Name
1	ItsyBitsy Pin 26
2	ItsyBitsy Pin 32
3	ItsyBitsy Pin 25
4	ItsyBitsy Pin 29
5	ItsyBitsy Pin 32

SSD13306

GND VCC SCL SDA
1 2 3 4

Num	Name	ConnectTo
1	GND	ItsyBitsy Pin 32
2	VCC	ItsyBitsy Pin 31
3	SCL	ItsyBitsy Pin 23
4	SDA	ItsyBitsy Pin 22

RGB LED

Num	Name	ConnectTo
1	Red	220 Ohm resistor->ItsyBitsy Pin 30
2	Grnd	ItsyBitsy Pin 32
3	Green	220 Ohm resistor->ItsyBitsy Pin 9
4	Blue	220 Ohm resistor->ItsyBitsy Pin 24

Datasheets

AdaFruit ItsyBitsy (32-bit ARM®, SAMD51 Cortex®-M4F MCU)

Data Sheet: *http://ww1.microchip.com/downloads/en/DeviceDoc/60001507E.pdf*

MICROCHIP - 25LC512-I/P - 512K SPI™ Bus Serial EEPROM DIP8, one primary one backup.

Data Sheet: *http://ww1.microchip.com/downloads/en/DeviceDoc/20005715A.pdf*

SSD1306 I2C LED display 128x32 pixels.

Data Sheet: *https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf*
https://www.vishay.com/docs/37894/oled128o032dlpp3n00000.pdf

Why PasswordPump?

https://siliconangle.com/2020/01/20/lastpass-suffers-outage-first-denied-quietly-confessed/

Why you shouldn't store passwords in your browser

Most web browsers offer to store your passwords for you. This might seem like an ideal way to keep track of your passwords – but it's actually a bad idea. Here are some reasons why:

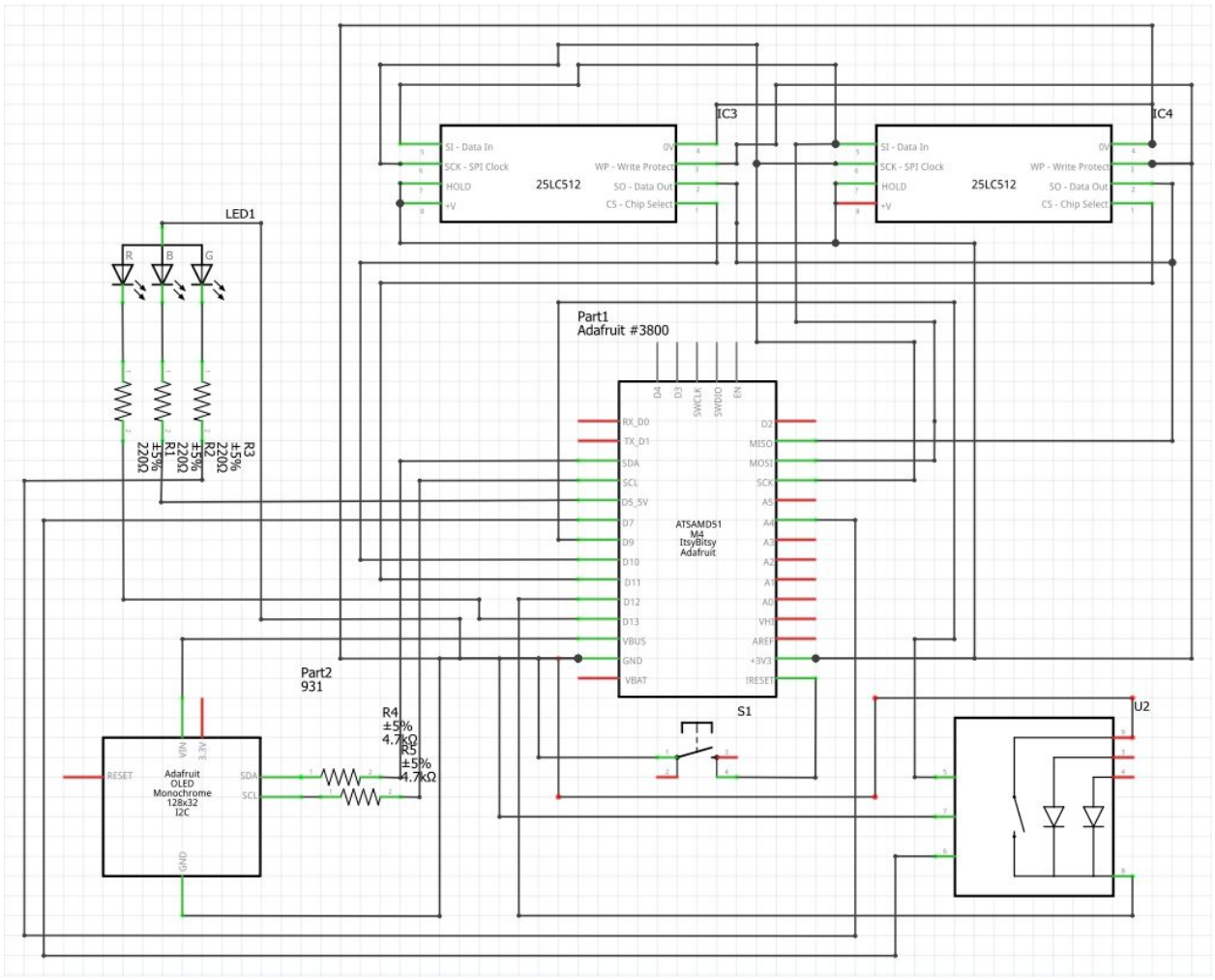
- The password security on browsers isn't that great – even if you are using a secure browser. Usually, these passwords are stored in plaintext. There are also tools available online that can give hackers access to your computer (either physically or remote access schemes) and view/steal passwords stored in the browser.
- Your browser will only record the username and password you enter into a web page. It won't help you generate a password, or tell you if the password is strong, or remind you that you already used this same password on 10 other pages.

From https://www.techspot.com/news/83704-best-password-managers.html:

How safe are password managers? Good discussion:
https://security.stackexchange.com/questions/45170/how-safe-are-password-managers-like-lastpass

---

Schematic


IC3
IC4
25LC512
SI - Data In
SCK - SPI Clock
HOLD
+V
WP - Write Protect
SO - Data Out
CS - Chip Select
LED1
R
B
G
R3
±5%
220Ω
R2
±5%
220Ω
R1
±5%
220Ω
Part1
Adafruit #3800
ATSAMD51
M4
ItsyBitsy
Adafruit
RX_D0
TX_D1
SDA
SCL
D5_5V
D7
D9
D10
D11
D12
D13
VBUS
GND
VBAT
D4
D3
SWCLK
SWDIO
EN
D2
MISO
MOSI
SCK
A5
A4
A3
A2
A1
A0
VHI
AREF
+3V3
!RESET
S1
Part2
931
R4
±5%
4.7kΩ
R5
±5%
4.7kΩ
Adafruit
OLED
Monochrome
128x32
I2C
RESET
VIN
3.3V
SDA
SCL
GND
U2