**Doorbot 2.0**

User Manual



**Table of Contents**

1.0 Introduction………………………………………………………………………………..2

2.0 Getting Started…………………………………………………………………………….2

2.1 Adding a user……………………………………………………………………..2

2.2 Deleting a user……………………………………………………………………3

2.3 Delete the database…………………………………………………………….3

2.4 Display a user……………………………………………………………………..3

2.5 Using an Existing Database………………………………………………….3

2.6 Saving a Database……………………………………………………………….3

2.7 Launching the Daemon……………………………………………………….4

3.0 Advanced Configuration………………………………………………………………4

3.1 Modifying ‘db.txt’ directly…………………………………………………...4

4.0 Hardware Layout…………………………………………………………………………4

4.1 Door circuit diagram…………………………………………………………..5

4.2 Embedded pinouts……………………………………………………………..6

**1.0 Introduction**

Doorbot 2.0 is an iButton authentication system for allowing authorized users access to a building. The system is designed to compare the code stored on an iButton key to a database of trusted users. If the user is present, Doorbot 2.0 will trigger a relay completing the circuit to the door lock motor. If the key is not recognized then the user will be denied access to the building.

**2.0 Getting Started**

The software driving Doorbot 2.0 can be in two states: daemon mode, where the software is scanning for iButton keys and admin mode, where the administrator can make changes to the list of authorized users. To start in admin mode enter *project –admin* in the command window while in the same directory as the project.

Important Note!

After modifying the database in admin mode, the daemon will need to be reset for the changes to take effect.

Once you have gotten past the prompt at the start of the configuration utility, you will be presented with a menu interface that has a prompt at the bottom. There will be a list of options down the side with numbers in front of them. To choose an option, simply enter the number into the prompt and press the return key.

**2.1 Adding a user**

To add a new user to the database enter “1” into the prompt at the bottom of the screen and press the return key. Fill in the appropriate fields and remember that both the first and last name fields are limited to 30 characters and the iButton value is seven numbers between 0 and 127.

**2.2 Deleting a user**

To Delete a user, enter “2” at the prompt and press the return key. Type in the user’s name at the next prompt and press the return key. If a match is found, the user will be deleted. If no match is found, a message will display telling you no match is found.

**2.3 Deleting the database**

To delete the entire database, enter “3” at the prompt and press the return key. A prompt will appear and ask if you are sure you want to proceed; enter yes to proceed. This will delete the entire database from memory. To make this permanent, save the changes to the database.

**2.4 Display a user**

To display the values of a specific user, enter “4” at the prompt and press the return key. Enter the user’s name at the next prompt and press the return key. If a match is found a record will be displayed. If not a message will appear to tell you that the record does not exist.

**2.5 Using an existing database**

To use an existing database on a file, enter “5” at the prompt and press the return key. At the next prompt, enter the path to the database file you wish to use. Please ensure that the database file is properly formatted or it will not be properly read (See 3.0). Once the database is loaded, you must save the database in order for the daemon to use it. It will overwrite the contents of db.txt to become the new database.

**2.6 Saving changes to the database**

To save changes to the database, enter ether “6” or “7” at the prompt and press the return key. “6” saves the database to ‘db.txt’ and “7” does the same thing but also exits the program.

**2.7 Displaying the entire database**

To display the entire database, enter “10” at the prompt and press the return key.

**2.8 Launching the daemon**

To launch the daemon, simply start the program without any arguments from the command line.

**3.0 Advanced Configuration**

Since Doorbot 2.0 stores it’s database in a text file, it is possible to modify this directly to add, modify or delete users.

**3.1 Modifying db.txt directly**

The database is stored in a file called db.txt and it is always located in the same directory as the executable. It has a specific format and breaking this format will corrupt the database. The format is:

*dd:dd:dd:dd:dd:dd:dd;firstname;lastname*

Where dd:dd:dd:dd:dd:dd:dd is the iButton address in 7 unsigned chars.

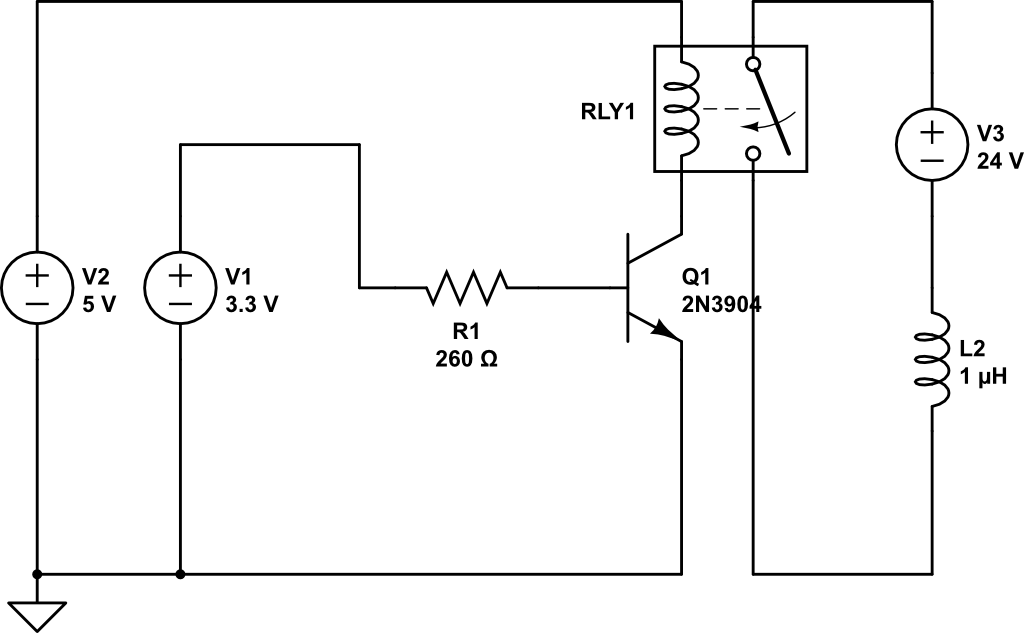
Example: 34:23:03:12:45:23:28 would be valid.

You can add, modify and delete any of these entries as long as the formatting is preserved. You also must restart the daemon before any changes come into effect.

**4.0 Hardware Configuration**

The door circuit works by triggering a relay that acts as a switch for the deadbolt motor circuit. The main circuit uses a +5v source from a GPIO pin on the Raspberry Pi embedded platform. A transistor hooked up to a +3.3V pin on an Arduino Uno prototype board controls this circuit. When the door event is tripped, 3.3V is applied to the base of the transistor and the triggering circuit is closed.

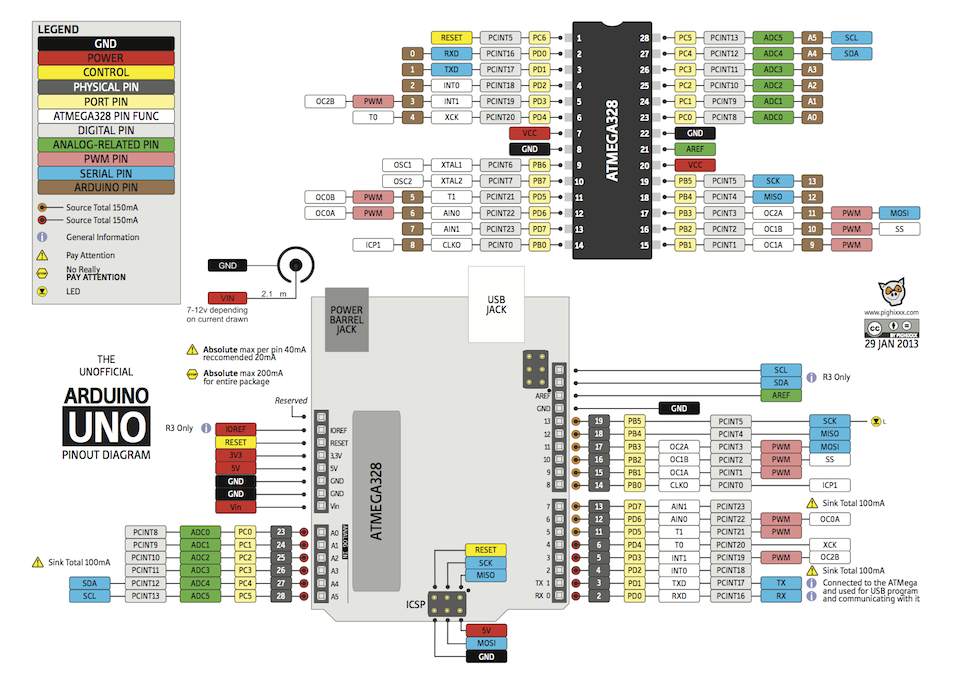
**4.1 Door Circuit Diagram**



/\* Yes, It’s missing the caps Sean. Stop nitpicking! \*/

**4.2 Embedded Pinouts**

**** 🡨 The GPIO pins on the Raspberry Pi.

****

The pinout diagram from the Arduino Uno.