**Doorbot 2.0**

User Manual



**Table of Contents**

1.0 Introduction

2.0 Getting Started

2.1 Adding a user

2.2 Deleting a user

2.3 Delete the database

2.4 Display a user

2.5 Using an Existing Database

2.6 Saving a Database

2.7 Launching the Daemon

3.0 Advanced Configuration

3.1 Modifying ‘db.txt’ directly

4.0 Hardware Layout

4.1 Door circuit diagram

4.2 Embedded pinouts

**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.

Important Note!

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

**2.1 Adding a user**

**2.2 Deleting a user**

**2.3 Deleting the database**

**2.4 Display a user**

**2.5 Using an existing database**

**2.6 Saving changes to the database**

**2.7 Launching the daemon**

**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:

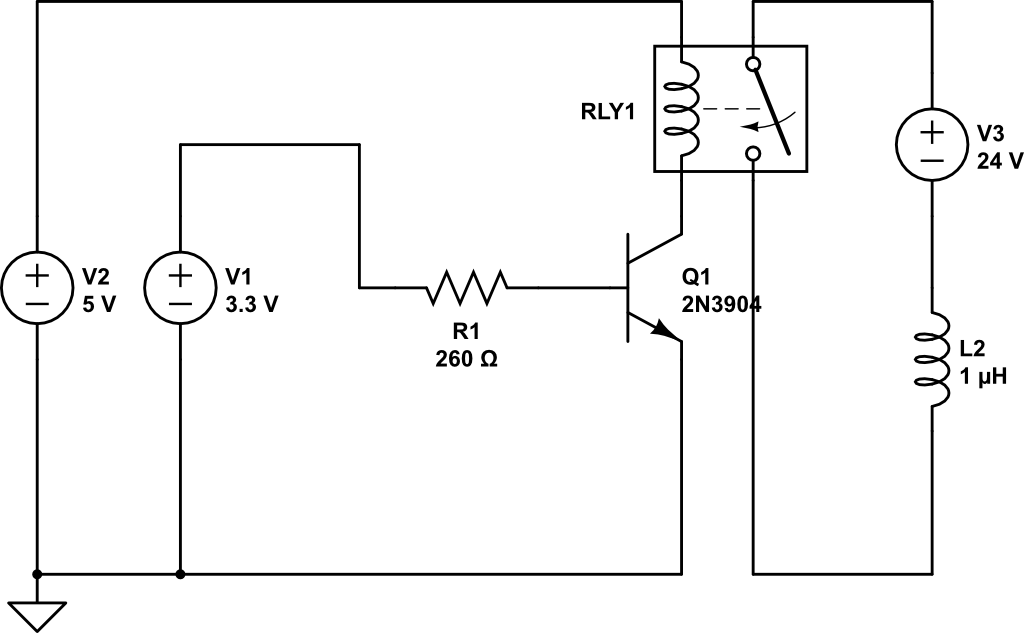
*iButtonKey;firstname;lastname*

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.

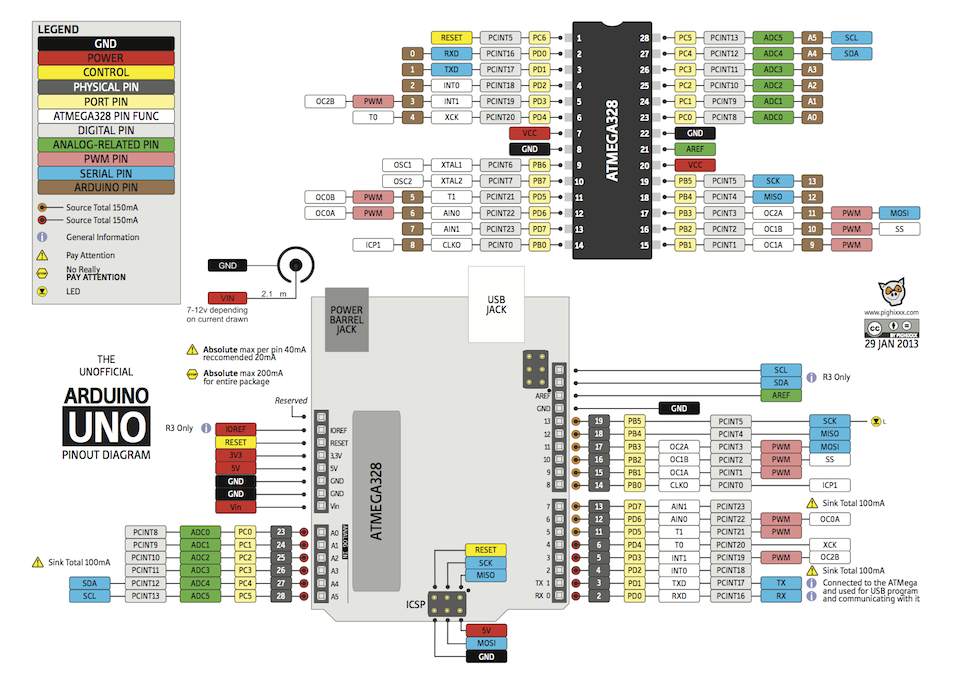
**4.1 Door Circuit Diagram**



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.