



# Electronic Lock Method

## 1. Arduino

The Arduino authenticates a 4-digit passcode supplied via USB/Serial.

### 1.1. Arduino

- a. Open the Arduino package
- b. *[Optional]* Download the sample code for the included sensors from Elegoo:  
<http://www.elegoo.com/download/>  
select “Elegoo UNO R3 Project The Most Complete Starter Kit”
- c. Connect your laptop to the lab Wifi:  
Network: TP-LINK\_C73C2E  
Password: 81558490

### 1.2. Test the Arduino (Blink! No Blink!)

- a. If you do not have the Arduino IDE installed, download and install<sup>1</sup>
- b. Plug in the USB cable to the computer and Arduino board
- c. Run the Arduino IDE
- d. Load the Blink example: File -> Examples -> Basics -> Blink
- e. Tell the computer to talk to the board using the right port: Tools -> Port -> choose the port that has the Arduino UNO on it
- f. Click the right-arrow ( $\Rightarrow$ ) button – this compiles the program and uploads it to the board
- g. After a few seconds, there should be a blinking orange LED on your board :)
- h. Now load the BareMinimum example (it does nothing): File -> Examples
- i. Compile and upload – your blinking LED should no longer blink.

### 1.3. Build the Lock Circuit

- a. See the provided diagram, be cautious of...
  - Resistor values (one is unlike the other three); orientation doesn't matter
  - Capacitor orientation (minus stripe/short wire = negative)
  - LED orientation (short wire = negative)
  - Button orientation (both legs should be on the same side of the divide)
  - Motor wires (black=negative, red=positive, white=control via Arduino)
  - Shorts (any time positive/negative meet without something in the middle)
- b. Have your circuit inspected before supplying power/moving on

### 1.4. Program the Arduino with Lock Software

- a. Download Zip: <https://github.com/natederbinsky/htmaa>
- b. Download Zip: <https://github.com/joshmarinacci/CmdArduino>

---

<sup>1</sup> <https://www.arduino.cc>

- c. Unzip htmaa-master.zip, open unit3/server/htmaa/htmaa.ino (in Arduino IDE)
- d. Use the CmdArduino library: Sketch->Include Library->Add .ZIP Library
  - Choose CmdArduino-master.zip
- e. Compile & Upload

## 1.5. Command the Arduino!

- a. Invoke the Serial Monitor in the Arduino IDE
  - Either upper right corner, magnifying glass OR Tools->Serial Monitor
- b. Make sure to choose “Carriage return” and “9600 baud” in the two dropdowns
- c. Type “move 1234” (no quotes) – the motor should move and the green LED blinks
- d. Type “move 4321” – the red LED should blink (bad passcode!)
- e. Type “change 1234 4321” – the blue LED should blink (passcode changed!)
- f. Type “move 4321” – green should blink and motor should engage
- g. Click the button on the board – the blue LED should blink (passcode is now 1234)

## 2. Raspberry Pi (RPi)

The RPi provides a client interface to the Arduino via a Python web-server site.

### 2.1. Install RPi in Case

- a. Bottom really matters, top not so much
- b. Label the RPi with the hostname!

### 2.2. Install Raspbian (the RPi Operating System)

- a. Insert the supplied MicroSD card
- b. Plug in monitor, keyboard, mouse
- c. Plug in power
- d. Allow the RPi to boot, choose Raspbian, let it install (takes ~20 minutes)
- e. Reboot

### 2.3. Connect to WiFi

- a. In the taskbar (at the top), click WiFi
  - Use the provided ssid/password

### 2.4. Configure the RPi

- a. From the taskbar, click the Terminal launcher (monitor with black screen)
  - **sudo raspi-config**
- b. Change User Password (2)
- c. Advanced (7) -> Hostname (A2) – use the label!
- d. Advanced (7) -> SSH (A4)
- e. Finish, Reboot

### 2.5. Update RPi Software

- a. Launch Terminal
  - **sudo apt-get update**
  - *[Optional]* **sudo apt-get upgrade**
- b. Reboot

## 2.6. Install VNC for Remote Desktop

- a. Launch Terminal
  - **sudo apt-get install tightvncserver**
  - **vncserver**
- b. Set a password (can be different than SSH)

## 2.7. Connect Remotely to the RPi

- a. On another computer, use an SSH program
  - Windows: putty
  - Mac/Linux: Terminal: **ssh pi@hostname.local**
  - Host: hostname.local
  - User: pi
  - Port: 22
  - Password: from 2.4c above
- b. Start the VNC Server
  - **vncserver**
- c. Use VNC client software
  - Windows: VNC Viewer
  - Mac: Chicken of the VNC
  - Host: hostname.local
  - Port: 5901 (display 1)

## 2.8. Install Webserver on the RPi

- a. Via Terminal in VNC, or SSH install the “flask” Python library
  - **sudo pip install flask**
- b. On the RPi, grab a copy of HTMAA software
  - **git clone https://github.com/natederbinsky/htmaa.git**

## 2.9. Run Webserver on the RPi

- a. Connect the Arduino to the power adaptor or to the 9V battery. Both options provided in the box.
- b. Remove the USB cable connection between the Arduino and the PC.
- c. Connect the RPi and the Arduino via USB.
- d. Via Terminal in VNC, or SSH, change to the “client” directory
  - **cd htmaa/unit3/client**
- e. Run the HTMAA web server program
  - **python web.py**
- f. [Optional] To shut down the RPi: ctrl+c to exit python, choose shutdown from the top-left menu

## 2.10. Remote Control

- a. From a phone or computer connected to TP\_LINK WiFi
- b. In a web browser, access <http://hostname.local:8080>
  - The RPi, found at hostname.local, is “serving” a website on port 8080