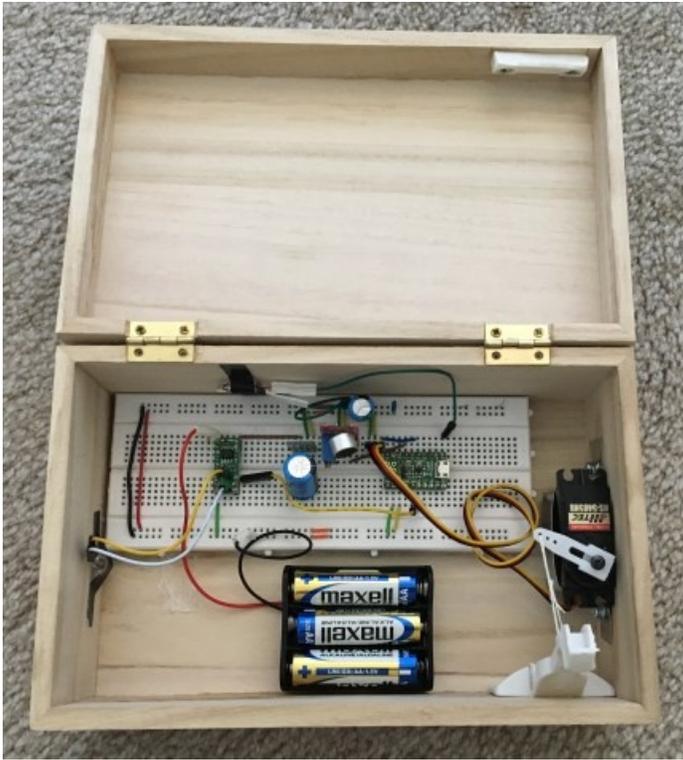


"Knock Knock Unlock" Puzzle Box

This is my version of a "knock knock unlock" puzzle box. A knock knock unlock puzzle box is a locked box that is opened by knocking on it with a particular sequence of knocks, for example "knock-pause-knock-knock-pause-knock-knock-knock" There are several versions of these boxes on the internet. For my version, I wanted to make a box that has nothing mounted on the outside -- no lights, no switches, no wires going through the box, and no outside latch. Just a blank box on the outside. Here are photos of and links to photos the inside and the outside of the box:



<https://www.keepandshare.com/doc6/18671/knock-knock-unlock-box-open-jpg-65k?da=y>



<https://www.keepandshare.com/doc6/18668/knock-knock-unlock-box-closed-jpg-50k?da=y>

PARTS

1. Mini Pushbutton Power Switch, LV (Pololu #2808) - 3.95
2. Mercury Tilt Switch, 1A @ 250V (Amazon) - \$2.37
3. Pololu 5V Step-Up/Step-Down voltage regulator (Pololu #2123) - \$5.95
4. Pololu A-Star 32U4 Micro microcontroller (Pololu #3101) - \$12.75
5. Anycubic Microphone Sound Detection Sensor Module (Amazon) - 6.99
6. Hitec 35485S HS-5485HB Digital Servo (Amazon) - 24.84
7. Micro Vibration Motor With Cable, 1.5V to 6V (Amazon) - \$8.69
8. 470 uF, 35VDC Capacitor (RadioShack #272-1030) - \$1.49
9. 1000 uF, 35VDC Capacitor (RadioShack #272-1032) - \$1.99
10. Safety 1st Spring-Loaded Cabinet & Drawer Latch (Walmart) - \$3.59
11. Pololu 3-AA Battery Holder (Pololu #142) - \$0.99
12. Breadboard (Sparkfun #PRT-12625) - \$5.95
13. Three Alkaline or Lithium size AA Batteries
14. ArtMinds Wooden Box, 8 1/2" L, 5 5/16" W, 3 9/16" H (Michaels crafts) - \$5.49

CONNECTIONS

Mini Pushbutton Power Switch

- VIN to 4.5V battery pack
- VOUT to voltage regulator, VIN
- A to mercury switch
- B to mercury switch
- OFF to 32U4, pin 12
- GND to common ground

Mercury switch

To A and B on the pushbutton power switch

(Note: The mercury switch is mounted vertically, with the tip down and the connection wires above.)

Voltage regulator

VIN to power switch, VOUT

VOUT to 5V common positive

GND to common ground

32U4 Micro microcontroller

5V to 5V common positive

GND to common ground

Pin 3 to microphone DO pin

Pin 5 to servo signal wire

Pin 9 to vibration motor red (+) wire

Pin 12 power switch OFF

Microphone

+ to 5V common positive

G to common ground

DO to 32U4 pin 3

Servo

Black to common ground

Red to 5V common positive

Signal wire to 32U4 pin 5

Vibration Motor

Red to 32U4 pin 9

Black to common ground

470 uF Capacitor

Positive to 5V common positive

Negative to common ground

1000 uF Capacitor

Positive to voltage regulator VIN

Negative to common ground

Cabinet & Drawer Latch

1. Trim the top of the latch off so it will fit in the box.
2. Drill a couple of 1/16" holes into the arm of the latch.
3. Loop heavy sewing thread through the 1/16" holes and the servo arm and tie the servo to the latch.

ARDUINO CODE

The following is just a very basic program to receive knocks and open and close

the box. You start the program by turning the box upside down, closing the mercury switch, and then turning the box upright. The sequence of knocks required to open the box is just four knocks within 30 seconds. Add your own code to make the required sequence of knocks more complex.

```
/*
```

```
This software was written by Bob Day.
```

```
It was written in May, 2016.
```

```
This software is licensed under the terms of the Creative Commons "Attribution Non-Commercial Share Alike" license, version 3.0, which grants the limited right to use or modify it NON-COMMERCIALY, so long as appropriate credit is given and derivative works are licensed under the IDENTICAL TERMS. For license details see
```

```
http://creativecommons.org/licenses/by-nc-sa/3.0/
```

```
This source code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
*/
```

```
#include <avr/sleep.h>
```

```
#include <avr/power.h>
```

```
#include <Servo.h>
```

```
// #include <SoftwareSerial.h>
```

```
// #include <serial.h>
```

```
// Pin assignments
```

```
static const int servo_control = 5;
```

```
static const int knock_interrupt_pin = 3;
```

```
static const int off_pin = 12;
```

```
static const int buzz_pin = 9;
```

```
// These values should be adjusted according to your needs
```

```
static const int OPEN_ANGLE = 120; // degrees
```

```
static const int CLOSED_ANGLE = 60; // degrees
```

```
int knocksToOpen = 4;
```

```
// The basic objects needed
```

```
Servo servo;
```

```
volatile long knock_heard = 0;
```

```
volatile long knockOkFlag = 0;
```

```
int knock_count = 0;
```

```
int delay_count = 0;
```

```
int idx; // Index variable.
```

```

int len;
int blinkLed = 13;

// The Arduino setup() function
void setup()
{
  pinMode(servo_control, OUTPUT);
  pinMode(3, INPUT);
  pinMode(off_pin, OUTPUT);
  pinMode(buzz_pin, OUTPUT);

  attachInterrupt(0, knock_detected, RISING);
  servo.attach(servo_control);
  delay(1000);

  servo.write(CLOSED_ANGLE);
  delay(1000);

  knock_count = 0;
  knock_heard = 0;
  knockOkFlag = 0;
  delay_count = 0;
} // End of setup().

void loop()
{
  if (knock_count >= knocksToOpen)
  {
    knock_count = 0;
    knock_heard = 0;
    knockOkFlag = 0;
    servo.write(OPEN_ANGLE);
    delay(7000);
    servo.write(CLOSED_ANGLE);
    delay(1000);
    digitalWrite(off_pin, HIGH); // Turn off the power.
  }
  else
  {
    if (knock_count < knocksToOpen)
    {
      servo.write(CLOSED_ANGLE);
      delay(1000);
    }
  }
}

```

```

while(1)
{
knockOkFlag = 1;          // Allow reception of knocks.
delay(250);
++delay_count;
if (delay_count >= 120)   // If 30 seconds have passed since starting,
{
digitalWrite(buzz_pin, HIGH); // send a long buzz and
delay(600);
digitalWrite(buzz_pin, LOW);
digitalWrite(off_pin, HIGH); // turn off the power.
}

knockOkFlag = 0;          // Knocks not allowed.
digitalWrite(blinkLed, LOW); // Turn off the blink LED.
if (knock_heard == 1)
{
digitalWrite(buzz_pin, HIGH); // Acknowledge the knock.
delay(250);                  // "
digitalWrite(buzz_pin, LOW); // "
knock_heard = 0;
++knock_count;
}

if (knock_count >= knocksToOpen)
{
break;
}
} // End of while(1).
} // End of loop().

//
// Functions
//

// Knock detected interrupt
void knock_detected()
{
if (knockOkFlag == 1)     // If knocks are OK,
{
knock_heard = 1;         // A knock was heard.
digitalWrite(blinkLed, HIGH); // Turn on the blink LED.
}

knockOkFlag = 0;          // Knocks are not OK for a while.
}

```

