Sean Hughes
Professor Hadas Kress-Gazit
MAE 3780
15 December 2020

The Pull-Up Automation Station

Design/Process Overview:

In the age of COVID-19, motivation has become more difficult than ever before. Especially during a global pandemic, where our personal health is of the utmost importance for both ourselves and our peers, mustering the daily drive to exercise has become exceedingly important, yet all the more challenging. The pull-up automation station's goal is to remedy this problem by providing users with an easy method to ensure proper pull-up form, adequate muscular tension, and a purely-human sense of motivation that makes working out fun—even in quarantine. This project has been designed to satisfy the home and office automation prompt, by way of automating something that is in dire need of help this year: user motivation.

This station functions first by sensing the user's hat with a color sensor positioned at a specific height, thus ensuring a full range-of-motion pull up with "chin over bar". If the user stays at the top of their pull-up for one second under muscular tension, then the Arduino's code will count the user's pull-up as good form and will reward them by dispensing candy for every set of 5. The difficulty of the device is modular, using a push button to alter the required number of pull-ups in a set from 5 to 10, therefore allowing the user to progress over time and challenge themselves if desired. Once a set of good pull ups has been counted, the circuit actuates a servo which rotates a dowel; the dowel has a hole drilled through its center such that once turned, it will expose a channel for candy to fall out of the structure with gravity, and into the user's hungry mouth. The rest of the mechanical system is designed to contain the candy and funnel it into the dowel-gate, while still being easily mountable at the top of a pull-up bar setup. The electronics are then mounted to the supports of the pull up bar to the left or right of the device using adhesive.

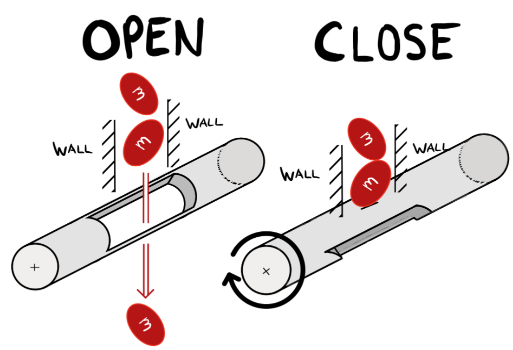


Figure 1: Dowel-Gate

This system not only allows users to be confident that their workout is effective, but also employs the user's natural desire for their favorite treats to motivate and energize a regular exercise routine—without the toxic diet-shaming of typical western workout motivation. Instead of prioritizing adherence to a spartan exercise regimen that is unsatisfying and harmful for the mental health of the majority of people, I wanted to create a device which user would *want* to use by designing an at-home workout which is both extremely effective[1] and easy to get motivated for. I was inspired by the mechanical design of gumball machines which have a similar rotating hatch as the dowel I designed, giving the project a small form factor and simple actuation requirements for the servo. I used scrap wood for the main body of the structure and a leftover mason jar as the container to stay under budget. I decided to use a micro position control servo instead of a continuous motor servo because it only needs to rotate ~90° to

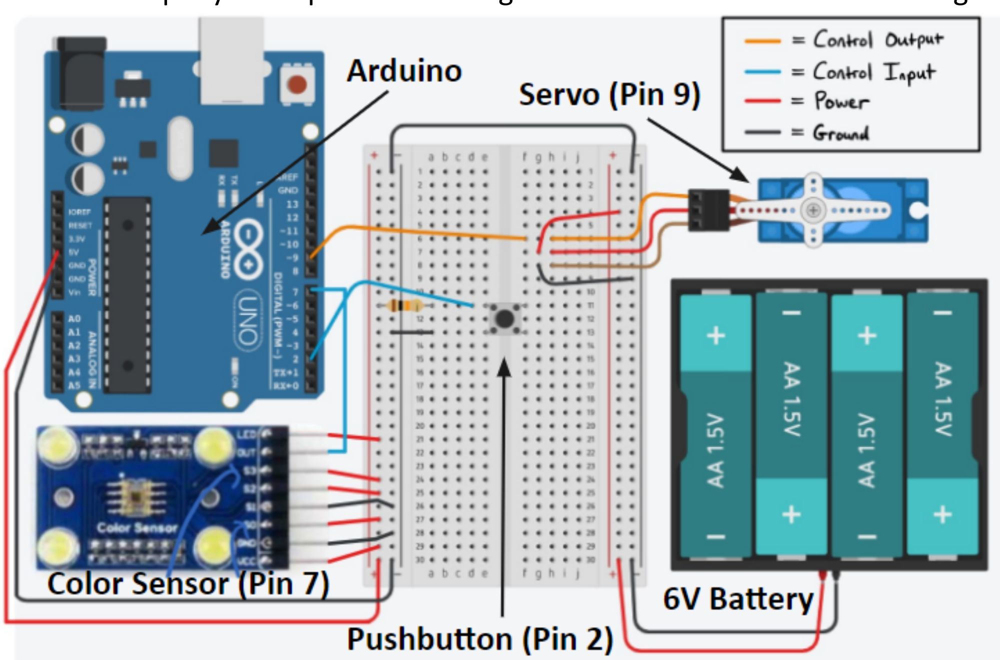
[1] Bodyweight resistance exercises have been demonstrated to promote the same muscle mass gains as traditional weightlifting, but only when proper form is enforced (Doma).

expose the channel in the dowel, and then rotate ~90° back in order to occlude the channel, stopping the flow of candy from the jar; thus the rotational limits inherent in this servo would actually help simplify the code. Additionally, I found it easiest to exploit the close proximity requirements of a color sensor in order to enforce proper form; the sensor will only detect the color of the user's hat if they reach the exact same height as the sensor. Each rep would therefore require perfect form and over time would train the user to have good form without external coaches, as well as actually enhancing their exercise efficiency through a "mind-muscle connection" (Schoenfeld). The arduino is powered using a portable phone charger since it is lightweight and more easily mountable than a laptop, in addition to powering the servo using a 6V battery pack for the same reasons. A push button is the simplest sensing option and does not further complicate the circuit, making it the most elegant solution for the user to choose difficulty. The entire system can be mounted using either velcro, super glue, nails, or adhesive tapes, with the latter being the most budget-friendly for my scenario; although, any of the listed options will do just fine as well, depending on what the user has on-hand in their home.

Bill of Materials:
All materials used for this project were common household objects, thus making the price of each item $0.

mason jar	scrap wood (2x4)	(kit) color sensor	(kit) 6V battery pack
superglue	duct tape	(kit) push button	(kit) Arduino Uno
scrap screws	leftover halloween candies	(kit) micro position control servo motor	(kit) breadboard, wires, & 10kΩ resistor

Circuit Schematic (Figure 2):
The color sensor is wired such that it will detect green color most accurately, as this is the color of the hat I use to keep my hair up while working out. The servo rotates the dowel-gate.



Code Description:

The code starts by setting the button and color sensor as inputs, while the servo is set as an output; external interrupts are enabled for the color sensor, and then it enters the while(1) loop. Here a function call gets the color from the color sensor through resetting all the timer 1 registers to desired values, and enabling interrupts for only 5ms. This triggers an external interrupt which can then measure the period of the wave detected by the color sensor. Going back to the while(1) loop, if this period is within the range pre-calibrated for the hat's color and then 1 second later the color sensor still detects the hat, then 1 is added to a pull-up count. Once this count is a multiple of 5 it will actuate a servo() function which resets the timer 1 registers to create a PWM for the servo to rotate as required for the dowel. If color is not being detected, then the while(1) loop polls the button's pin to see if it has been pushed. If pushed, then a toggle variable is set to 1, the count is reset, and now the while(1) loop will require the count variable to be a multiple of 10 (so that 10 pull ups are required for a reward now). Upon repressing the button, the count is reset again, and the toggle is set back to 0, therefore requiring 5 pull ups instead. A delay at the end ensures the sensors only check every 25ms.

Process Reflection:

The implementation of the dowel for this design was much more difficult than I initially assumed. The first issue that I encountered was that the hole in the dowel was too small for my candy to fall through without jamming in place and preventing further actuation. I solved this by using a small saw to form the hole into more of a crescent shape, such that the hole was much larger. Additionally, I found that once the candy exited the dowel and fell through, it simply dropped on the bottom of my wood casing and wouldn't roll out any further. To remedy this, I cut out the bottom of my wooden structure so the chute would go straight down, as I had originally planned to use a slide to propel the candy into the user's mouth, but soon realized that the frictional forces were too high to ensure smooth operation. My design therefore had evolved to simply have the candy fall directly into the mouth of the user.

Perhaps the most surprising issue was that my servo was not rotationally constrained, and I had completely not foreseen that I would have to secure the servo to a stationary object in my design. While it took some experimentation to constrain it with the materials I still had on-hand, I came up with a solution by super gluing two screws to the flat top of my housing, and then positioned the servo in between. The two screws therefore provide an equal and opposite torque to the servo, therefore allowing it to rotate the dowel upon actuation, rather than the entire servo unit rotating while the dowel stays in place (see: ***Fig. 3***). This was a fantastic learning experience for me personally, as I had come to realize that this is actually a fundamental part of nearly all designs including motors: the only way that a torque can be applied is if the motor is properly constrained itself. If I were to do this project differently, I would have cut the scrap wood in such a way that a baseplate could rotationally

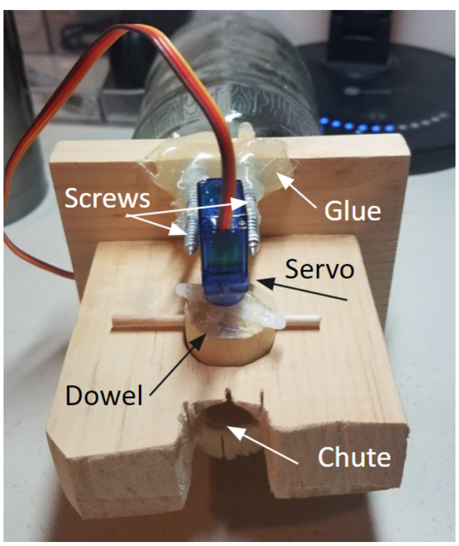


Figure 3: Servo Constraints

constrain the motor housing, without use of external screws or glue. Furthermore, I would want to design a more robust mounting mechanism for the entire project instead of simple adhesives (see: ***Fig. 4***). Ideally this would be removable so that users could take the structure down temporarily to refill the candy without needing a ladder. This could be accomplished by attaching velcro to the wall that the pull up bar is mounted to, as well as the backs of the electronics and the mechanical structure itself. However, the software-side design of the various sensors causing actuation worked without a hitch, and the color sensor was surprisingly accurate with whether or not it counted proper pull up form. In all of my tests it only miscounted once or twice, thus it would seem this code actually works quite well for the project, and shouldn't be altered. While the button performed just as reliably, if multiple people were using this device, I would add a second button to toggle the allowable color of the hat for the color sensor to detect. In this way, multiple users could simply toggle the preference to be for their specific hat.

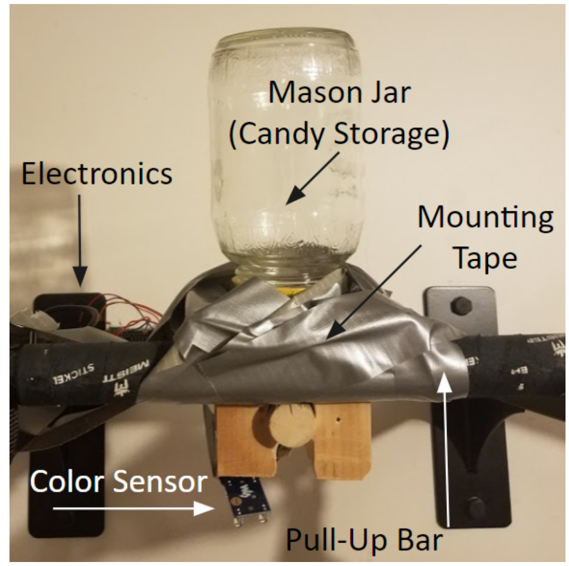


Figure 4: Mounted System

Feedback Reflection:

In the first milestone, my TA had suggested that I consider using an ultrasonic distance sensor instead of the color sensor, this way, the user would not have to use a specific hat each time they workout. This was actually quite helpful and really forced me to investigate whether or not my color sensor was the most efficient option for my project's functional requirements. In my testing, however, I found that the color sensor's close distance requirement allowed for pull-up form to be enforced much more reliably, and given that the color sensor could be calibrated to a "clear" filter, this setup could actually perform approximately the same requirements as the distance sensor could, if desired. Additionally, one of my teammates suggested that I should have a method to unjam the dowel if any candy ever gets stuck inside is. This critique was actually one of the inspirations for the updated crescent design of the dowel hole, as this allows for the force of gravity on the candy to tend to slide the candy into a more favorable position, rather than being force into a position against the wall where it might get stuck. One of the critiques of my design which wound up not being fairly problematic was that the method by which it is mounted is underdeveloped. As discussed previously in the project, numerous options are available to employ; although my use of adhesive tape was less convenient than other options for the user, it did not jeopardize the integrity of the design's functional requirements as initially implied by my TA. Despite this, his advice was still important to the consideration of more modular, user-friendly designs in the future if I were to redesign the project. Finally, one of the suggestions I had made to a different teammate was to utilize a continuous motor instead of the micro position servo that I had used in my own project. Her project seemed to utilize a larger range of motion for a door opening than mine did, and so I thought it would be much easier to implement practically than would the more restrictive motion of the mini servo.

Appendix

```
//Pull Up Automation Station Arduino Code:

//defining variables for modes & tracking
int count = 0;       //keeps track of pull-ups done in a set
int toggle = 0;      //toggles between normal mode (5 reps/candy) & hard mode (10 reps/candy)
volatile int period;  //period of the waveform detected by the color sensor
volatile int timer1;  //time when wave goes from high to low in ticks

ISR(INT1_vect)  // code that runs when external interrupt 1 is triggered
{
   if(PIND & 0b00001000) //if color sensor's output (pin 3) is high
   {
      TCNT1 = 0;       //reset the timer
   }
   else
   {
      timer1 = TCNT1;   //store current timer value
   }
}

void getColor()             //pings interrupt to sense color @ current moment
{
   //timer1 setup
   TCCR1A = 0b00000000;     //normal mode
   TCCR1B = 0b00000001;     //normal mode, prescaler = 1

   EIMSK |= 0b00000010;     //enable external interrupt for INT1 (pin 3)
   _delay_ms(5);            //delay to give interrupt time to trigger
   EIMSK &= 0b11111101;     //disable external interrupt for INT1 (pin 3)
   period = timer1*0.0625*2; //converting ticks to microseconds
}

void servo()  //actuates servo
{
   //timer 1 registers for PWM
   TCCR1A = 0b10000000;   //timer mode 8, clear OC1A on compare match (set output to low
level)
   TCCR1B = 0b00010010;   //timer mode 8, prescaler = 8
   ICR1 = 20000;        //setting TOP value to give the timer a period of 20 milliseconds (50Hz)
   OCR1A = 900;         //set compare register to start @ 900 in order to get pulse width = 0.9 ms

   while(OCR1A<2100){
```

```
        OCR1A++;         //starts off increasing OCR1A up to 2100 micro secs
        _delay_ms(2);
        }
    while(OCR1A>900){
        OCR1A--;         //after reaching 2100 microsecs, decreases OCR1A down to 900 micro secs
        _delay_ms(2);
        }
}

int main(void)
{
    Serial.begin(9600);
    DDRD = 0b11110011;     //setting button & color sensor as inputs into arduino (bit 3,2 = low)
    DDRB = 0b00000010;     //setting micro servo as an output for the arduino (bit 1 = high)

    //color sensor registers
    EICRA |= 0b00000100;  //enable external interrupt for any logical change on pin 3
    sei();              //enable interrupts globally

    while(1)
    {
      getColor();        //gets color from color sensor
      if(200 < period && period < 280)    //checking for green color (thresholds experimentally
determined)
      {
        _delay_ms(1000);  //wait 1 second under muscular tension
        if(200 < period && period < 280)  //checking if user is still under tension
        {
          count = count+1;  //marking that a good pull up was done
          if(count%5 == 0 && toggle == 0)   //if sensor measures 5 good pull ups & normal mode
activated
          {
            servo();  //actuate servo
          }
          if(count%10 == 0 && toggle == 1)  //if sensor measures 10 good pull ups & hard mode
activated
          {
            servo();  //actuate servo
          }
        }
      }
      else  //if color is not being detected then instead poll for button press
      {
        if(PIND & 0b00000100){ //check if button is not pushed
```

```
         }   //do nothing
         else                //check if button is pushed
         {
            toggle ^= 1;      //toggles from normal mode (0) to hard mode (1) or vice versa
            count = 0;        //resets pull-up count to avoid miscounts
         }
      }
      _delay_ms(10);          //check color only every 0.25s
   }
}
```

Works Cited

Doma, K., Deakin, G., & Ness, K. (2013, September). Kinematic and electromyographic comparisons between chin-ups and lat-pull down exercises. Retrieved December 17, 2020, from https://pubmed.ncbi.nlm.nih.gov/24245055/

Schoenfeld, B. J. (2018, March). Differential effects of attentional focus strategies during long-term resistance training. Retrieved December 17, 2020, from https://pubmed.ncbi.nlm.nih.gov/29533715/