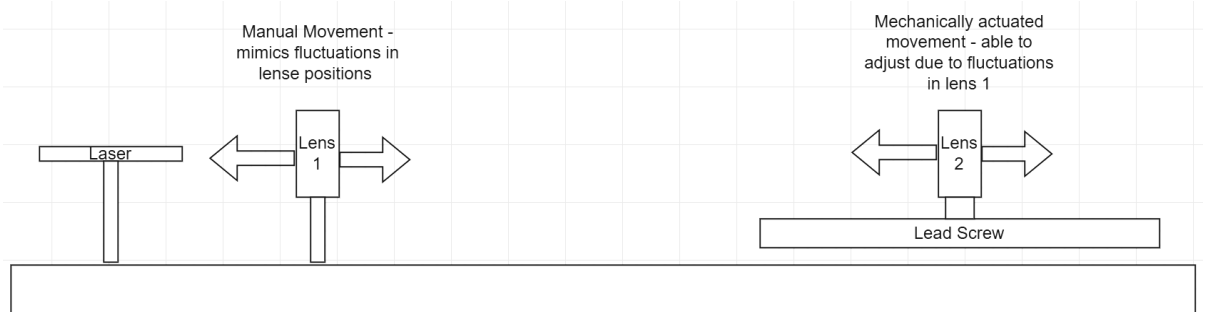
Motorized Optical Translation Stage
Benjamin Capinski & Karla Morales De Leon

Opportunity

One of the most time-consuming aspects of research when working with lasers is ensuring proper laser beam alignment and collimation. This process, normally done by hand, can be automated through the use of linear motorized translation stages. Unfortunately, solutions from conventional photonics suppliers can cost thousands upon thousands of dollars, with the cost rising for each additional centimeter of travel distance. We aim to make precision control of laser beam paths cheaper than ever and make motorized translation stages ubiquitous on optical tables worldwide by slashing that cost by several orders of magnitude. We will do that by sourcing significantly cheaper components, sacrificing a small amount of precision, accuracy, and repeatability in exchange for a much more affordable product.

Strategy

To demonstrate our translation stage, we have incorporated it into a beam expander that increases the red dot of a laser pointer by fivefold. Effectively a telescope, our setup uses a static 35mm lens to focus the laser beam, followed by a motorized 175mm lens to re-collimate the beam at its increased size. If these lenses are not precisely 210mm apart, the beam will undesirably diverge or converge. The latter lens position can be moved using a potentiometer in order to maintain this separation if the first lens moves.



In lieu of an optics table to supply rigidity, we have attached the whole assembly, including the laser and optics, to segments of aluminum T-slot framing. Our initial goal was at least 6 inches of travel distance and roughly 0.5 inches/second of translation speed. Our final construction surpassed this comfortably, at approximately ~10 inches of travel distance and a maximum of 1.5 inches/second of translation speed.

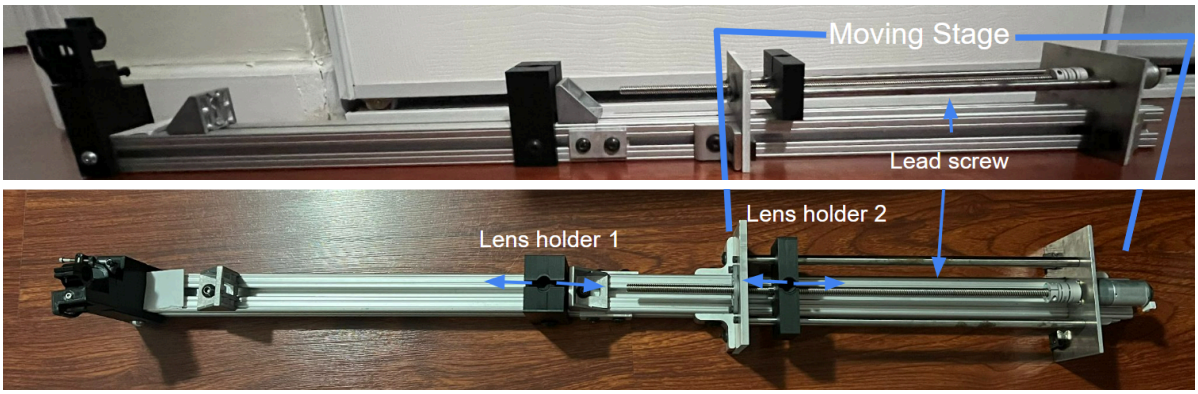


Fig. 1: From left to right; the laser mount, lens holder 1 (static), and lens holder 2 (motorized).

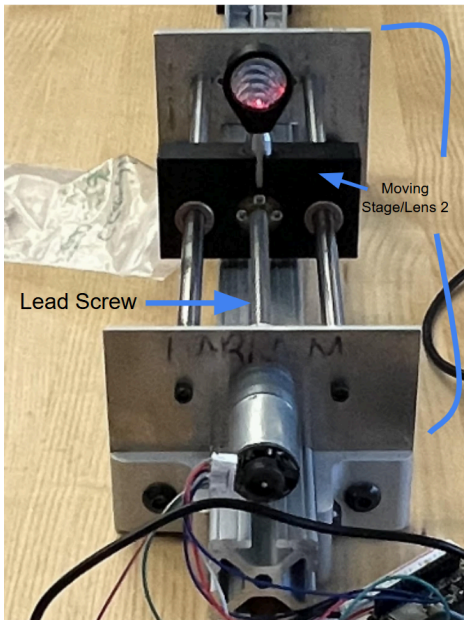


Fig. 2: An axial view of the rail, showcasing the translation stage itself. Note the lead screw, guide rails, and 3D-printed lens holder. A flexible shaft coupler is hidden by the bottom plate.

Functional-Critical Decisions & Calculations

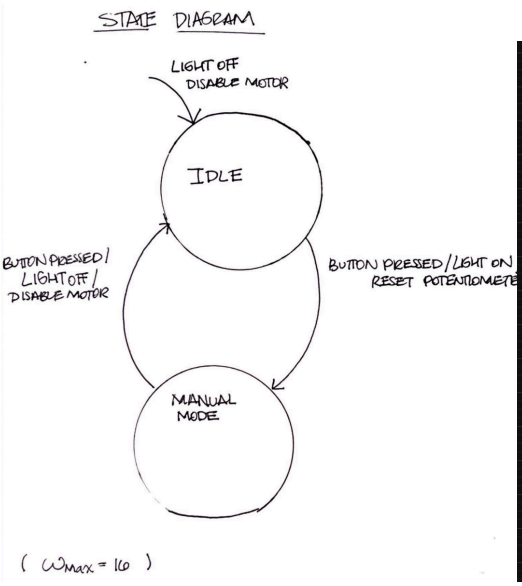
The bulk of our calculations focused on the non-mechanical aspects of our project, such as simulating the laser beam path and its intensity at certain points. For the translation stage, we did a back-of-the-envelope calculation to verify that the torque required was easily satisfied by our motor. Unfortunately, this is only the torque required when the system is not binding and fighting itself. That resistance is difficult to calculate and since we had the parts on hand, we just tested it *in situ*. It worked.

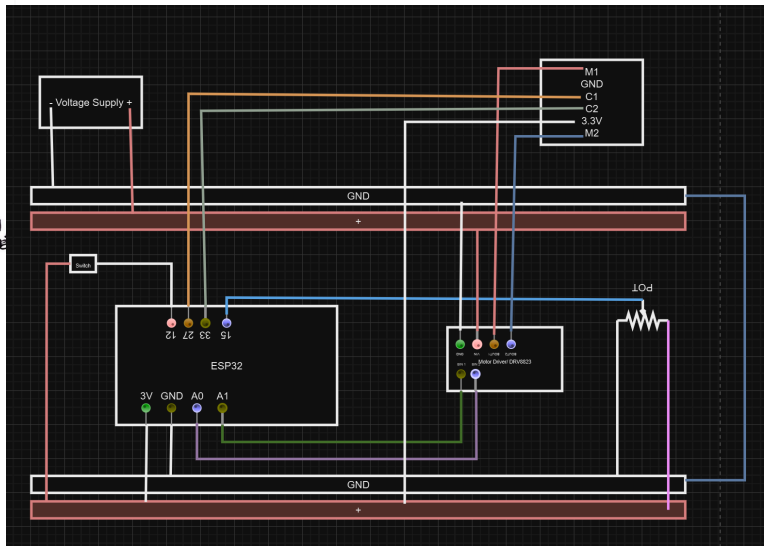
$$\tau = \frac{F \cdot L}{2\pi\eta} = \frac{(0.4\ N)*(0.01\ m)}{2\pi \cdot 0.5} = 0.00127\ Nm$$

Where η is the efficiency of a lead screw (between 0.2 and 0.8), about 50% for ours, L is the lead, so about 10 millimeters = 0.01 meters, and the force F is the weight of the stage carriage times the friction coefficient. F = mgμ = 100 grams * 9.81 meters per second squared * 0.4 ≃ 0.4 Newtons. Excellent machining tolerances, alignment, and structure rigidity are crucial for approaching this ideal torque value.

Circuit & State Transition Diagrams

Fig 3: Circuit diagram made w/ draw.io & our state diagram.





Reflection

What worked well - leveraging each other's strengths by delegating programming, CAD, and machining tasks to the appropriate team member. What we wished we did differently - start everything sooner, from machining to programming. This would've allowed us to achieve the full complexity we planned for.

APPENDIX

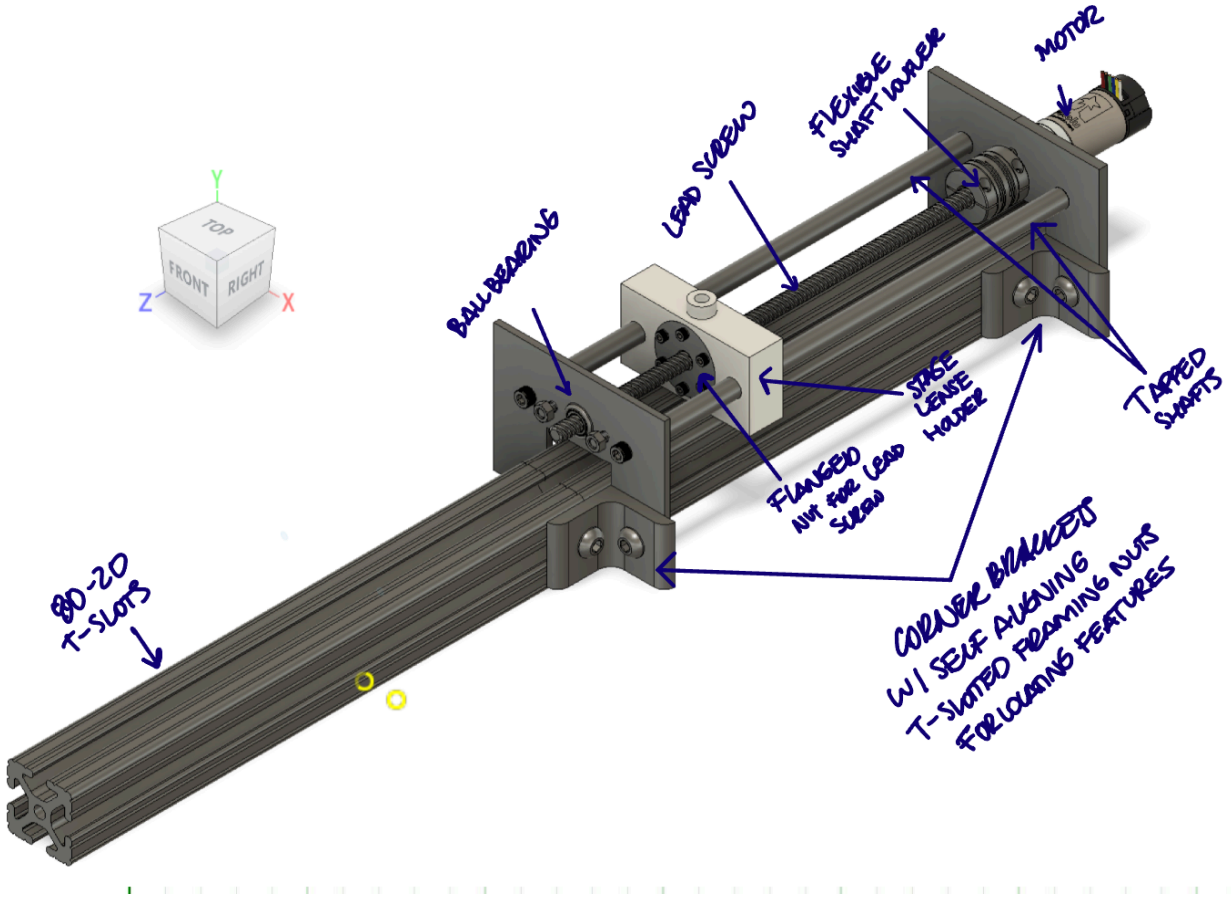
Bill of Materials

The overwhelming majority of our parts were scavenged from salvage bins in the physics and mechanical engineering buildings, or lent to us by the machine shop. The following BOM represents our best-faith representation of the parts we used, although we were unable to find model numbers for some parts.

Part Name	**Quantity**	**Cost per tem**	**Total cost**
635nm 5mW red laser (for safety during display)	1 (1 min.)	52	52
1.5” Aluminum T-slot - 3ft	1 (1)	34.40	0 (found)
1” lens kit (f = 35 to 175mm)	1	0	0
Photodiode	1	0	0
All the following components were sourced/CAD modeled from McMaster-Carr			
6208K492_Clamping Precision Flexible Shaft Coupling	1	0	0
4316N121_Tapped Linear Motion Shaft	1	0	0
8020-1515 X 12	2	0	0
90044A254_Black-Oxide Alloy Steel Socket Head Screw	1		
91274A127_Zinc-Flake-Coated Alloy Steel Socket Head Screw (mounting shafts)	4	0	0
91239A420_Button Head Hex Drive Screw	6	0	0
2391N24_Fast-Travel Ultra-Precision Lead Screw	1	0	0
4316N121_Tapped Linear Motion Shaft	2	0	0
25d-metal-gearmotor-20.4-encoder	1	0	0
94209A619_Thread-Forming Screws for Soft Metal (mounting motor)	2	0	0
Ball Bearing Holder (3D Printed, will utilize ME Machine Shop)	1	0	0
6208K492_Clamping Precision Flexible Shaft Coupling	1	0	0
47065T845_Silver Corner Bracket	3	0	0
90591A260_Zinc-Plated Steel Hex Nut	2	0	0
Stage Lense Holder (3D Printed)	1	0	0

2391N11_Flange Nut with M8 x 2.50 mm Thread for Fast-Travel Ultra-Precision Lead Screw	1	0	0
4473N18_Mounted Ball Bearing with Two-Bolt Flange	1	0	0
90044A255_Black-Oxide Alloy Steel Socket Head Screw (To attach Flang Nut for lead screw)	6	0	0
End piece to connect Shafts (3D Printed, will utilize ME Machine Shop)	1	0	0
2391B11 Flange Nut	1	0	0
PLA Filament (OVERTURE PLA)	1	15.21	15.21
Cable Connector (AMAZON)	1	10.46	10.46
Calipers (VINCA)	1	21.94	21.94
Total Spent:			~$97

CAD Screenshot



Code

This is the code we used for the showcase. It implements manual control of the translation stage.

```
#include <ESP32Encoder.h>
#define BIN_1 26    // direction
#define BIN_2 25    // duty
#define LED_PIN 13  // built-in LED
#define POT 15      // trimpot
#define PD 14       // photodiode
#define BTN 32      // button
#define IDLE 1      // idle state
#define MANUAL 2    // manual state

// Setup variables
volatile bool buttonIsPressed = false;
int state = IDLE;
int pot_low_threshold = 1500;
int pot_high_threshold = 2500;
int pot_value = 2000;

hw_timer_t* debounce_timer = NULL;
int debounceDelay = 500;

// Open loop control variables
const int freq = 5000;
const int resolution = 8;
const int MAX_PWM_VOLTAGE = 240;

// Functions to be called when interrupts are triggered
void IRAM_ATTR button_isr() {
  if (timerReadMilis(debounce_timer) > debounceDelay) {
    buttonIsPressed = true;
    timerRestart(debounce_timer);
  }
}

void setup() {
  pinMode(BTN, INPUT);
  pinMode(POT, INPUT);
  pinMode(LED_PIN, OUTPUT);
  attachInterrupt(BTN, button_isr, CHANGE); // TODO maybe change to FALLING

  ledcAttach(BIN_1, freq, resolution);
  ledcAttach(BIN_2, freq, resolution);

  debounce_timer = timerBegin(1000000);
  Serial.begin(115200);

  // Initialize
  led_off();
  disable_motor();
}
```

```
void loop() {

  switch (state) {
    case IDLE:
      if (CheckForButtonPress()) {
        led_on();
        reset_pot();
        state = MANUAL;
      }
      break;

    case MANUAL:
      if (CheckForButtonPress()) {
        disable_motor();
        led_off();
        state = IDLE;
      } else {
        set_duty();
      }
      break;
  }
}

bool CheckForButtonPress() {
  if (buttonIsPressed == true) {
    buttonIsPressed = false;
    return true;
  } else {
    return false;
  }
}

void reset_pot() {
  pot_value = analogRead(POT);
  pot_low_threshold = pot_value - 250;
  pot_high_threshold = pot_value + 250;
}

void set_duty() {
  pot_value = analogRead(POT);
  if (pot_value > pot_high_threshold) {
      ledcWrite(BIN_1, LOW);
      ledcWrite(BIN_2, MAX_PWM_VOLTAGE);
    } else if (pot_value < pot_low_threshold) {
      ledcWrite(BIN_2, LOW);
      ledcWrite(BIN_1, MAX_PWM_VOLTAGE);
    } else {
      disable_motor();
    }
}
```

```
void disable_motor() {
  ledcWrite(BIN_1, LOW);
  ledcWrite(BIN_2, LOW);
}

void led_on() {
  digitalWrite(LED_PIN, HIGH);
}

void led_off() {
  digitalWrite(LED_PIN, LOW);
}
```