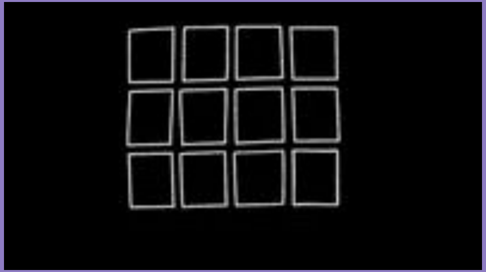
Modeling Particle Collisions on a Macroscopic Level

By: Logan Barrus, Phil Coady, Aassik Pazhani, Javier Tort, Jason Vazquez

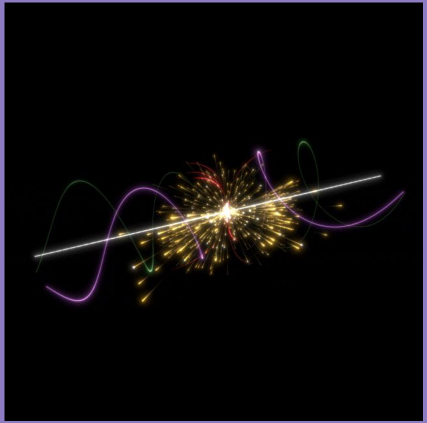
What is the purpose of particle physics?

- Studying fundamental particles and forces of nature
- Exploring dark matter and dark energy
- Testing of theoretical models such as:
  - Supersymmetry
  - Extra Dimensions
  - String Theory
- This field is also very useful for its production of new technology which is driving research in other fields of science research



What is the purpose of particle collisions?

- Identification of Particles and their Interactions
  - Behavior and properties of subatomic particles
  - Studying the fundamental forces that bind these particles together (the strong & weak force)
- Testing and Developing Models
  - Supersymmetry, Extra Dimensions, String Theory
  - Prediction testing
  - Potential new discoveries of non-theorized phenomena



Modeling particle collision on a macro-scale

Quantum Scale

- Small (hard to see)
- Complex
- Difficult to observe directly
- Results are hard to interpret

Macroscopic Scale

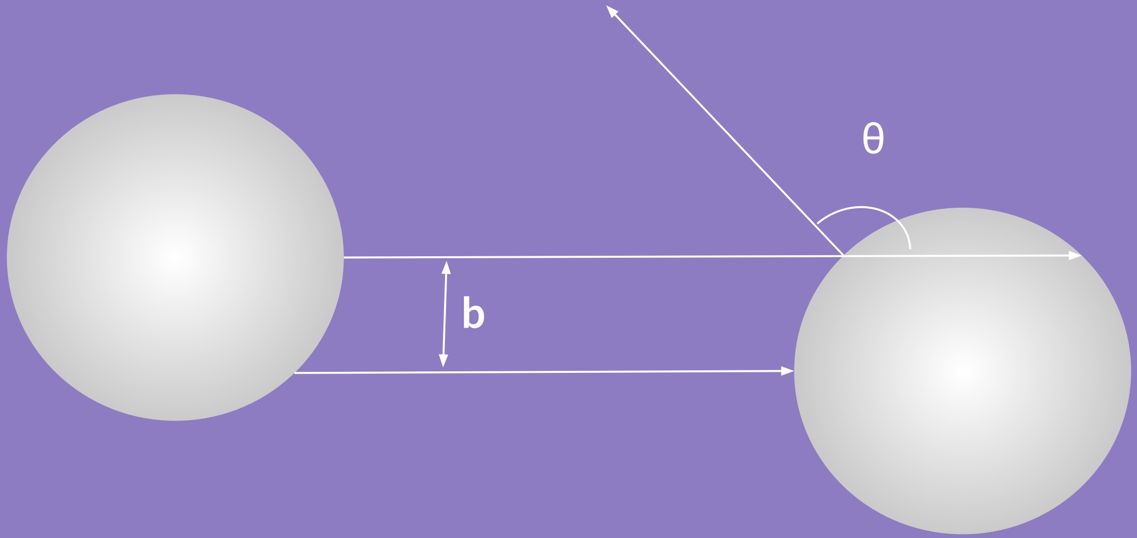
- Big (easy to see)
- Simpler
- Can be seen and recorded with a camera
- Results are much easier to digest

Bottom Line

A macroscopic model is an effective tool for exploring new ideas in the particle physics field. These models can then be used to guide the design of experiments that will test these ideas.

What do we need to model these collisions?

- Scattering angle: $\theta = 2tan^{-1}(\frac{b}{v_1 + v_2})$

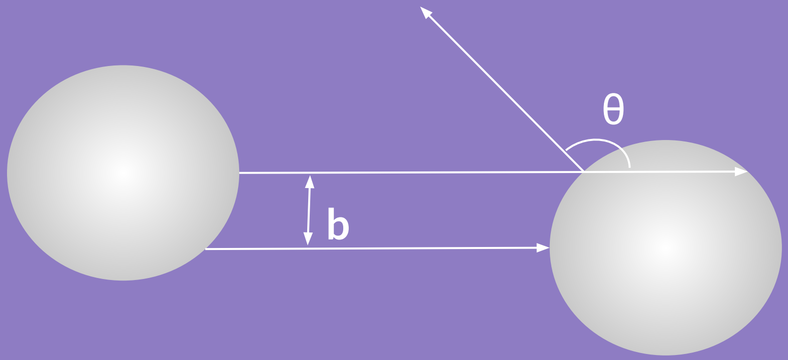


What do we need to model these collisions?

- Final Velocity of each puck:

$$v_1^f = \frac{(m_1 - m_2)}{(m_1 + m_2)} v_1^i + \frac{2m_2}{(m_1 + m_2)} v_2^i cos(\theta)$$

$$v_2^f = \frac{(m_2 - m_1)}{(m_1 + m_2)} v_2^i + \frac{2m_1}{(m_1 + m_2)} v_1^i cos(\theta)$$



How do we obtain these measurements?

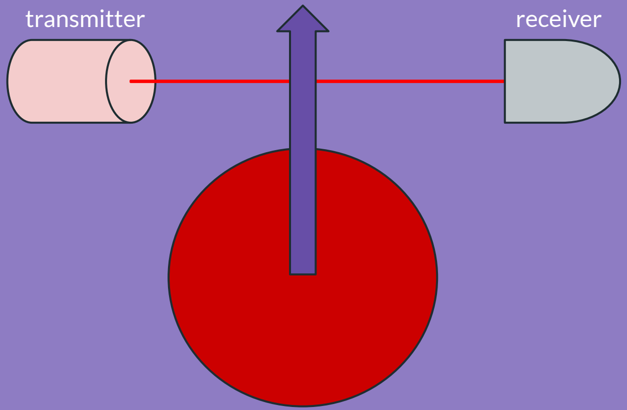
- We will be using the following instruments for data taking:
  - Laser Transmitter and Laser Sensor Receiver
    - This will give initial velocity of the pucks
  - ArduCAM Mini
    - This will record a video of the collisions that will be used to plot the trajectories
  - Small Reduction Stepper Motor
    - This will give us control over the impact parameter ‘b’

Laser Transmitter and Laser Sensor Receiver

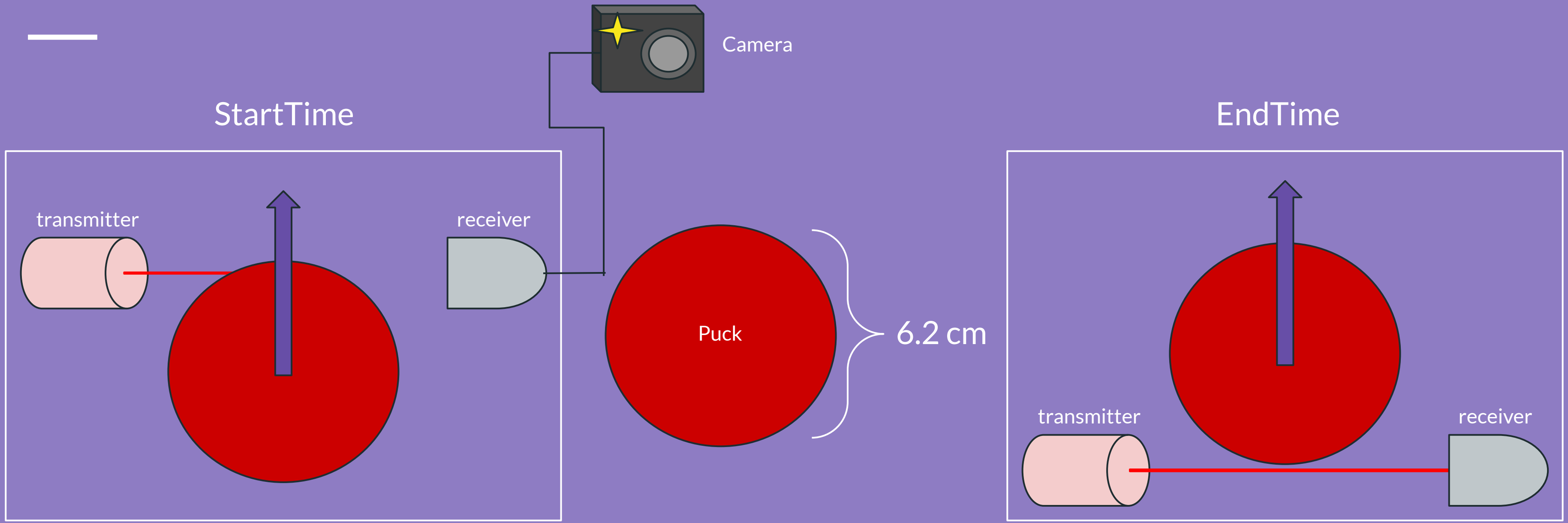
- Two Functions
  - Trigger arducam camera to start video
  - Measure initial velocity
    - Saved to CSV on SD card which we process later

```
if (value1==1) {
  if (check1 == 1) {
    endTime1 = millis();
    check1 = 0;
    Serial.println("this is sensor1");
    Serial.println(endTime1);
    Serial.println(startTime1);
    Serial.println(endTime1 - startTime1);
    myFile.println((diameter/(endTime1 - startTime1))* 1000);
```

V = x/t

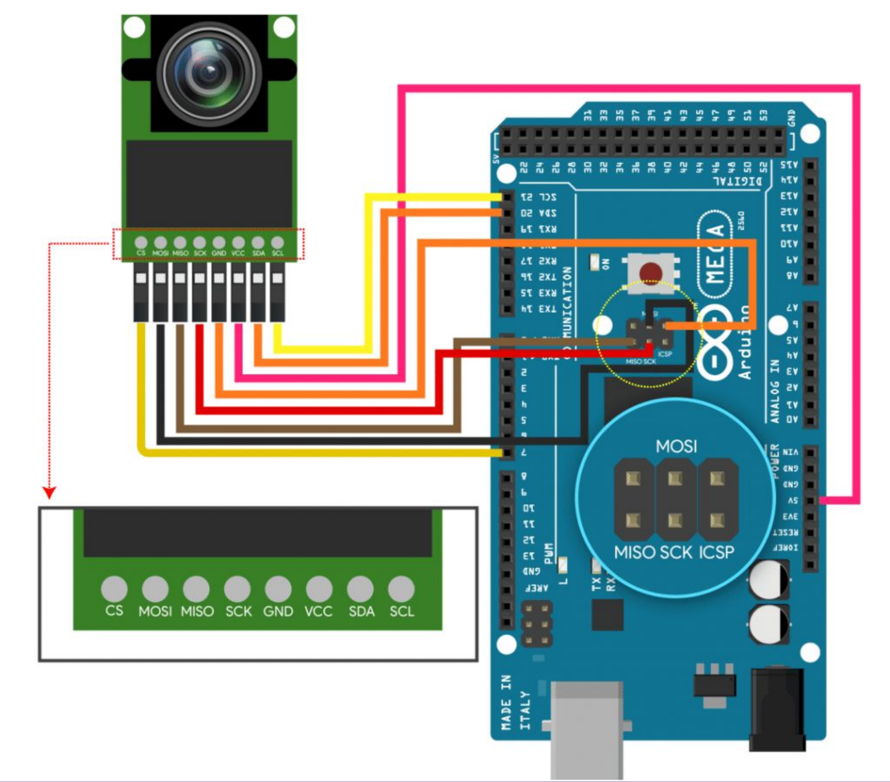


Laser Transmitter and Laser Sensor Receiver



Initial Velocity = Diameter of Puck [6.2 cm] / (StartTime - EndTime)

ArduCAM Mini



ArduCAM Mini

Set resolution, total frames, and pin number:

```
#define SERIAL_SPEED 115200
#define BUFFSIZE 512  // 512 is a good buffer size for SD writing. 4096 would be better, on boards with enough RAM (not Arduino Uno of course)
#define FRAME_SIZE OV2640_160x120
#define WIDTH_1 0xA0 // Video width in pixel, hex. Here we set 320 (Big Endian: 320 = 0x01 0x40 -> 0x40 0x01). For 640: 0x80
#define WIDTH_2 0x00 // For 640: 0x02
#define HEIGHT_1 0x78 // 240 pixels height (0x00 0xF0 -> 0xF0 0x00). For 480: 0xE0
#define HEIGHT_2 0x00 // For 480: 0x01
#define FPS 0x1E // 15 FPS. Placeholder: will be overwritten at runtime based upon real FPS attained
#define TOTAL_FRAMES 100  // Number of frames to be recorded. If < 256, easier to recognize in header (for manual hex debug)
//set pin 7 as the slave select for SPI:
#define SPI_CS  7
// SD card Select pin:
#define SD_CS 53  // 9 on Arducam adapter Uno and SD shields
```

Set hardware typer (camera model):

```
//Step 1: select the hardware platform, only one at a time
#define OV2640_MINI_2MP
//#define OV3640_MINI_3MP
//#define OV5642_MINI_5MP
```

ArduCAM Mini

Camera Trigger:

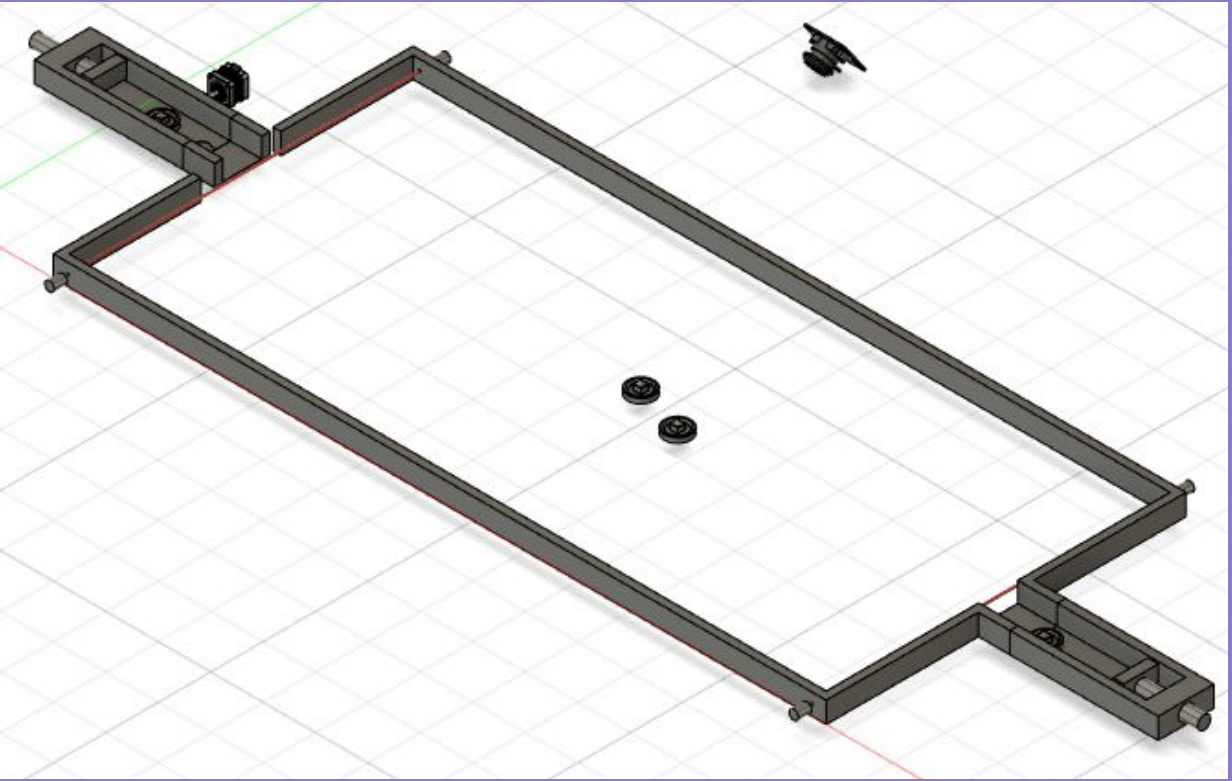
(left)

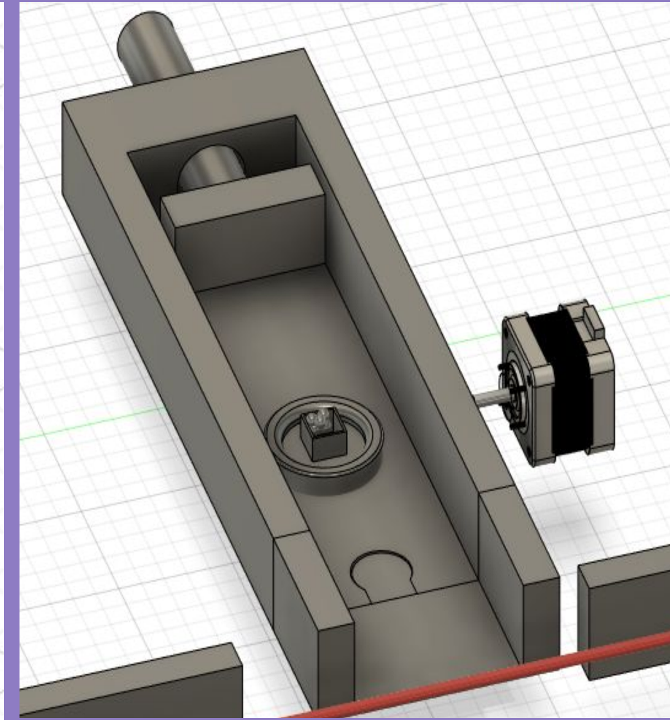
```
void loop() {
  digitalWrite (LaserPin1, HIGH);
  digitalWrite (LaserPin2, HIGH);
  bool value1 = digitalRead(sensor1);
  bool value2 = digitalRead(sensor2);
  if (value1==0){
    if (check1 == 0) {
      startTime1 = millis();
      check1 = 1;
      Video2SD();
    }
  }
  if (value1==1) {
    if (check1 == 1) {
      endTime1 = millis();
      check1 = 0;
      Serial.println("this is sensor1");
      Serial.println(endTime1);
      Serial.println(startTime1);
      Serial.println(endTime1 - startTime1);
      //myFile.println((.062/(endTime1 - startTime1))* 1000);
      //myFile.print("\n");
    }
  }
```

```
  //Write video data, frame by frame
  for ( frame_cnt = 0; frame_cnt < TOTAL_FRAMES; frame_cnt++)
  {
#if defined (ESP8266)
    yield();
#endif
    temp_last = 0; temp = 0;
    //Capture a frame
    //Flush the FIFO
    myCAM.flush_fifo();
    //Clear the capture done flag
    myCAM.clear_fifo_flag();
    //Start capture
    myCAM.start_capture();
    // Wait for frame ready
```

Inside Video2SD() disregarding everything related to SD card and figuring out the picture types:
(right)

So how do we create such a thing?





Possible Additions (if we have time)

- Live machine-learning using the Arduino Nano
- Live TCL Display
- Quark-Gluon Plasma Model:
  - using a cage of polyester beads attached to the air hockey pucks, concepts such as conservation of momentum, behavior of granular materials, and elastic collisions could be understood better. Understanding this would be helpful for gaining an intuition into how quark-gluon plasmas interact in a collision.
- Electromagnetic Interaction:
  - using small pieces of foam (think tiny pool noodles) protruding from this aforementioned cage, one could potentially model the electromagnetic influence of one particle onto another. For example, even if the impact parameter ‘b’ is so large that the two particles don’t collide, they could still interact with each other through their own electric fields. It would be intriguing to provide a visual demonstration of this concept.

What do we do with the data we've recorded?

By the end of our collision shots, there should be three sets of data that will need to be analyzed:

- Laser Transmitter and Laser Sensor Receiver
    - Taking the .csv file, we will use python to process the velocity measurements and relate them to the scattering angle
- Stepper Motor
    - Also writing to .csv, this will give us the other variable needed to calculate the scattering angle
- ArduCAM Mini
    - Using TensorFlow, we will attempt to create a model that will predict the trajectories based on the original video files we upload (along with the other data values we've collected)

Thanks for Listening