Team 516: Instrumented Baseball

Operation Manual

David Adams, Mathew Brown, Riley Ferrer, Yanni Giannareas, Charles Whitaker

FAMU-FSU College of Engineering

Table of Contents

1. Project Overview
   1. Description
   2. Objective
   3. Key goals
   4. Assumptions
2. System Description
   1. Piezoelectric sensors
   2. Bluetooth Communication
   3. Lithium-Ion Batteries
   4. 3D-printed elements
   5. Molding of the baseball
3. System Integration
4. Operation of the Device
5. Troubleshooting for Common Issues
   1. Battery
   2. Connections
   3. Sensors
   4. Molding
   5. Housing
6. **Overview**

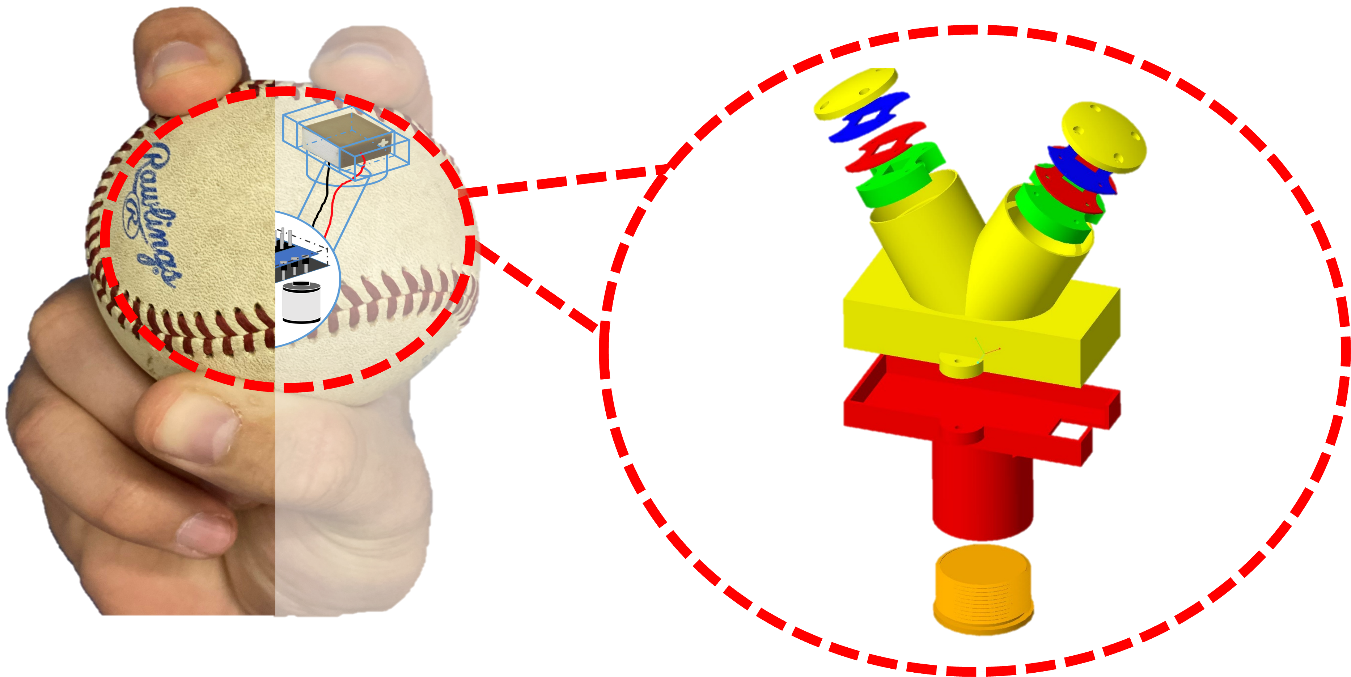


Figure 1: Instrumented Baseball design

* 1. **Project Description**

The game of baseball is changing rapidly in the modern era. What used to be a sport based on instinct has now become a game revolved around numbers. Professional teams spend time and money focusing on the analytical aspects of the game to seek an edge against competition. In today’s era of the sport, the pitcher’s motion, baseball speed, and pitch path are all measures tracked during games and practices. Our project involves modifying a baseball to find another process that allows teams to base their game decisions on.

When a pitcher throws a baseball, a dynamic force is created at each fingertip. This dynamic force is the combination of a compressive and shear forces. With this instrumented baseball, we aim to create an accurate way of measuring these forces applied when throwing the ball. This is carried out by using specialized electrical parts placed within the ball where the user’s fingertips would be placed when throwing the ball.

* 1. **Project Objective**

The objective of this project is to implement piezoelectric sensors to determine a pitcher’s dynamic fingertip force as a pitch is thrown. Within doing so, the group aims to create 3D printed housings in which all components can be secured within the baseball. Lastly, it is important that the instrumented baseball is on par with size and weight regulations.

* 1. **Key Goals**

Key goals within the scope of the project are listed below:

* The instrumented baseball has a moment of inertia close to that of a regulation baseball.
* The instrumented baseball accurately senses an applied load (90 mph pitch should be roughly 117 N).
* The instrumented baseball can plot time-dependent data and determine the point of maximum dynamic force when the ball is being thrown.
* The supplied voltage for all electrical components falls within a 5-12 V range.
  1. **Assumptions**

There were several key assumptions that were made as this project progressed. First, the baseball needs to be caught by a net. The group calculated there to be a very large impact force if the ball were to be caught in a catcher’s glove. To relieve issues with our sensors breaking it needs to be assumed that the pitch will be thrown into a net. It is also important to assume that these sensors will be placed below in a 4-seam fastball manner. This is the most common pitch among pitchers and the group decided the instrumented baseball would be used extensively if the ball were to be tailored to this pitch. Lastly, it is also assumed that the user will have access to a device that can connect to Bluetooth. The group aims to send sensor data through a Bluetooth module in which the user can access the data through a Bluetooth device.

1. **Component/Module Description**
   1. **Piezoelectric sensors**

The piezoelectric chips used for this project are the TA0505D024 pressure sensors and the PL5FB shear sensors that can be found on Thorlabs website. Sensors are to be assembled in a stack that is composed of one pressure sensor and two shear sensors to capture the three components of the force applied.

The function of these piezoelectric sensors is to generate a voltage whenever a force is applied on top of them. Each sensor has a positive and negative electrode through which the voltage is generated. Given the size of these sensors, certain aspects regarding their mechanical and electrical handling must be considered. Some of these aspects are as follows:

* Do not store these at temperatures above 80°C.
* Avoid soldering directly to the electrodes.
* Consider using a resistor (>1kΩ) connected between the electrodes to avoid discharging.
* Do not load the sensors at locations close to the edges to avoid failure.

Figure 2 below shows the location of the electrodes for both the pressure and shear sensors. Since these electrodes are flat, there are two recommended ways in which cables can be attached to them. The first method is to make mechanical contact between the electrode and a copper wire and then wrap the connection with the use of electrical tape. This assures that the mechanical contact is maintained during operation. This method is primarily intended for individual testing of the sensors, as the addition of the tape can interfere with the force gathering.

Diagram

Description automatically generatedDiagram

Description automatically generated

Figure 2: Location of the electrodes for pressure (left) and shear sensors (right)

The second method involves the use of copper tape. A reliable electrical connection can be achieved by sticking the tape into the electrode from one side, and into a cable from the other. It is recommended that the copper tape is wrapped around the cable and secured using insulation to prevent current leakage. This method is preferable for the final assembly. Figure 3 shows how copper tape can be used in shear sensors. It is extremely important that, when making the connection between the tape and the electrodes, the two pieces of tape do not touch at all.

A picture containing wooden, metalware, wood

Description automatically generated

Figure 3: Copper tape method for electrical connection in shear sensors.

To measure the analog response from the sensors, the Arduino Blue Nano microcontroller will be used. It is small enough to fit inside the baseball and has enough capacity to read the six sensors used. The function used to read the data is analogRead(), which converts the analog input voltage from the sensor to a digital value between 0 and 1023. The following code can be used to verify the response of each individual sensor, assuming the positive end is connected to the A0 pin and the negative to the GND pin:

Graphical user interface, text, application, email

Description automatically generated

The equation that is used to convert the reading into force was found by extensive calibration of the sensors using a load frame and a micrometer stage. This code can be expanded to all six sensors by declaring pins A0 to A5 as input pins and reading their analog values separately. Since the Bluno Nano only has 2 GND pins, we recommend that each sensor stack uses a common ground for convenience. This can be achieved with the use of heat shrink butt connectors, where the three ground cables can be attached to a single cable that goes to the GND pin. The overall circuit diagram can be seen in Figure 4 below.

Graphical user interface

Description automatically generated with low confidence

Figure 4: Circuit diagram of the Bluno Nano.

* 1. **Bluetooth Incorporation**

The group plans to use an Arduino Blue Nano microcontroller for several reasons. One is because of its small, sub-compact design the microcontroller can easily fit inside the instrumented baseball and weighs only 5 grams. The microcontroller also has a built-in Bluetooth module in which information can be transmitted to Bluetooth devices. Lastly, the microcontroller has enough analog pins (7) in which the piezo sensors can be connected to. Each sensor stack needs 3 analog connections, so the group can get a better picture of this Arduino microcontroller below in figure 3.

A picture containing text, electronics, circuit

Description automatically generated

Figure 3: Arduino blue nano microcontroller

Operating the Bluetooth module requires 3 main steps in which the user needs to connect their device to this Bluetooth device. First, the user needs to find the settings menu on their device and make sure their Bluetooth is ON. Secondly, the user must download the “Bluetooth for Arduino” application and connect to the devices named “Bluno” to establish communication with the microcontroller. Lastly, the user should wait 30 seconds before the first throw to ensure that the sensors are fully decompressed and provide reasonable data. An illustration of a typical Bluetooth window can be seen below in Figure 5.

A screenshot of a computer

Description automatically generated with low confidence

Figure 5: Device’s Bluetooth window showing real-time data of sensors

* 1. **Lithium-Ion Battery**

The Arduino Blue Nano microcontroller needs at least 9V and 19 mA of current to sufficiently power the Bluetooth controller over a period. Due to the sub-compact scale of the project, the group chose to implement a stack of 3V lithium-ion batteries. Stacking 3 batteries in series will allow for the proper voltage (9V).

A picture containing indoor

Description automatically generated

Figure 5: Battery stack hooked up to

This battery stack can easily be assembled and can determined from the following steps:

* + 1. Stack three 3V batteries on top of one another (positive end connects to negative end of another battery)
    2. Attach a wire lead to the top of stack (positive end) and another wire lead to the bottom of the stack (negative end)
    3. To ensure secure connections wrap the battery stack in electrical tap
    4. Connect positive lead to V­in on the Arduino microcontroller
    5. Connect negative lead to GND on the Arduino microcontroller
    6. Once all leads have been fastened to the headers on the Arduino board, check the light on the board
       1. If light is NOT on (red) then check connections (see figure 5 above)
       2. If light is ON you have successfully connected the battery stack to the Arduino

This wired battery stack will be placed in the red housing section shown in figure 1. This cylindrical tube will allow the user to easily place the stack of batteries inside. The orange end cap (seen in figure 1) is screwed into the battery housing and fastens the stack so that the components do not come loose.

* 1. **3D-Printed Elements**

The group decided to 3d-print all housing components found within the baseball. This design process was crafted within PTC-Creo 8 and is broken down into 6 primary components found within the appendix. For the software used to 3d print, the group used open-source slicing program Cura. Within Cura the printer was set for each component on .12 quality to ensure a precision print as well as a 30% infill to ensure the parts are sturdy enough to be within ball and not break when thrown. We also are printing using standard PLA as its sturdy enough and can be printed safely in a closed environment. Drawings and dimensions of these components can be found in the Appendix.

* 1. **Molding of the Baseball**

When molding the baseball, the group decided on a silicon compound sealant as the material of choice. This sealant is the “DAP Ultra Clear Silicon Sealant” as its properties can match the requirements needed for a baseball. This material once cured, has a similar texture to that of the rubber cage balls used for training. With properties such as being waterproof, crackproof, and minimal flexibility, this silicon compound is what is ideal regarding material selection. Typical baseballs fall within the density of where the sealant is and that increase makes up for some of the material carved out of the ball.

The molding portion is separated into these steps:

The resin is squeezed out using a caulk gun into the 3D printed mold as shown in the figure below.

While wearing gloves and in a well-ventilated space the silicon compound is pressed into the mold using your hand to ensure as much air bubbles are popped, and the ball is filled tight.

Grab the main housing piece and press within the mold to the edge of sphere

Fill the other half using steps 1-2

Clamp both halves together and let cure for 24-48 hours in a ventilated area.

Diagram

Description automatically generated Figure 6: 3D printed mold pieces

1. **Systems Integration**

The following steps must be followed when trying to incorporate all the systems of the instrumented baseball into a single, fully functional device:

1. Using copper tape, make an electrical connection between the electrodes of the sensors and a set of cables.
2. Locate the sensors in their appropriate spot of the sensor stack housing.
   1. Make sure to mark which cables correspond to the positive and negative electrodes of each sensor.
   2. Avoid any type of contact between different electrodes or different sensors.
3. Place each sensor stack in their appropriate location in the middle housing.
4. Fasten the sensors into the stack using a set of 4 bolts and nuts.
5. Pass the cables through the slots provided in the housing.
6. Solder the resistors to the positive and negative electrode cables of each sensor.
7. Using a heat shrink butt connector, connect the three negative electrode cables of each sensor stack into a single cable.
8. Connect each cable to the appropriate pins as indicated by the circuit diagram.
9. Place the Arduino into the bottom housing.
10. Fasten the bottom and middle housing using a set of 2 bolts and nuts.
11. Fasten the middle housing and the sensor stacks with the use of epoxy.
12. Connect the set of batteries to the Arduino using the Vin and GND pins.
13. Place the set of batteries on top of the bottom cap
14. Fasten the batteries by clamping the bottom cap through the threaded section of the bottom housing.
15. Place the integrated housing into the top part of the baseball mold and secure it by covering it up with the bottom part of the mold.
16. **Operation of Device**

The following steps must be sequentially followed when operating the device:

1. Open the “Bluetooth for Arduino” app on your phone.
2. In the list of devices, search for the one named “Bluno” and click on it to connect with the microcontroller.
3. Select the option of “Serial Monitor” to send commands and view outputs from the Arduino.
4. After Bluetooth communication is achieved, start gripping the ball while contacting the indications for each finger.
   1. It is recommended to not put too much pressure on the ball at this point to avoid faulty data to be gathered.
   2. The indications assume a right-handed throw. If the throw is to be performed with the left hand, place your finger in the opposite locations and interpret the data accordingly.
5. Throw the ball into a non-destructive cover, such as a net or a blanket.
6. Retrieve the ball from the net or blanket.
7. The data values are gathered by the microcontroller and updated to the user every 2 seconds.
   1. Data values include the three force components (shear in X, shear in Y, pressure) for each fingertip.
8. Leave the device to rest for at least 2 minutes.
   1. This is done to relax the charge that is accumulated by the sensors, avoiding failure or incorrect readings later.
9. After rest, the ball is ready to be thrown again.
10. **Troubleshooting for Common Issues**
11. **Battery**
    * 1. Device does not turn on
         1. Double check batteries are in the correct position
         2. Recheck battery stack is connected properly
         3. If the device still does not turn on replace with fresh batteries
12. **Connections** 
    * 1. Sensors are working but no signal to the Arduino
         1. Check for wire tears and broken wires
         2. Check the quality of the copper foil between the sensors
      2. No signal to the display device
         1. Move closer to the baseball
         2. Check the Arduino is working properly
13. **Sensors**
    * 1. Sensor not transmitting signal
         1. Check that sensor is not damaged
         2. Check that the copper foil is properly connected
         3. Check connection to Arduino
14. **Molding**
    * 1. If molding appears to have indents
         1. Check mold
         2. Fill with proper amount of silicone
      2. Mold will not print properly
         1. Recalibrate 3D printer
         2. Check filament quality
         3. Check CAD drawings are to the right dimensions
15. **Housing**
    * 1. Housing bolt holes does not line up
         1. Check 3D printer calibration
         2. Check filament quality
         3. Check print quality
         4. Check infill
         5. Check CAD drawing dimension
      2. Housing pieces do not line up with other pieces
         1. Check 3D printer calibration
         2. Check filament quality
         3. Check print quality
         4. Check infill
         5. Check CAD drawing dimension

**Appendix - Drawings**

