# OLA 12

## IT Technology - ITTA1

## Team B2


Authors:

*Aleksis Kairiss*
alka28793@edu.ucl.dk

*Alexander Bjørk Andersen*
aban28821@edu.ucl.dk

*Dainty Lyka Bihay Olsen*
dlbo28887@edu.ucl.dk

*Daniel Rasmussen*
dara28918@edu.ucl.dk

*Jacob Suurballe Petersen*
jspe28926@edu.ucl.dk

*Nikolaj Hult-Christensen*
nihu28952@edu.ucl.dk

*Sheriff Dibba*
shdi28860@edu.ucl.dk


University College

11 / 01 / 2021 - Character count: 16.897

# Table of Contents


**Introduction** **2**
- Project overview 2
- Goals 2

**Inventory** **2**

**Project plan** **3**

**Teamwork organization** **3**

**Use cases** **3**

**Block diagram** **4**

**Recreation manual** **5**

**Documentation of system tests** **5**

**User manual** **6**

**Final product and casing** **7**
- Barrel attachment 7
  - Printed parts 7
  - Electrical components 8
- Handguard rail attachment 8
  - Printed parts 8
  - Electrical components 8
- “Scope” attachment 9
  - Printed parts 9
  - Electrical components 9

**Codeflow** **9**

**Thingspeak** **11**

**Improving and expanding upon the product** **11**

**Conclusion** **12**

# Introduction

This is the semester project part 2 for ITT1 where the students will define their own IoT system from the building block they have learned about in Part 1.

## Project overview

The project is for nerf gun enthusiasts and/or for kids who like to play around with nerf and BB guns. The user would be able to have statistics like one can in modern computer games. A display counter will show the user how many darts are left in the magazine. Another thing that the operator will be able to do, is to know many darts have been fired in total using thingspeak.

## Goals

The primary goal of this project will be to create a working model and have it attached to a nerf gun.
The secondary goal would be to have a working model attached to a BB gun.

# Inventory

System:
- Raspberry pi zero W

Materials:
- 3D Printed Parts
- 9V battery

Sensors:
- Magazine Sensor - Hall sensor
  - Custom 3144 C8011 - Linear hall sensor
- Muzzle Sensors - IR sensors
  - Receiver: tcrt5000, Transmitter: LIR034

Peripherals:
- 7-segment display
  - 3461BS-1
- Buzzer
  - Active piezo buzzer

Other components:
- DC-to-DC converter
  - LTM8067Y (BOB)

# Project plan

The project plan section is where we have a clear overview of our project. The purpose of this is to organize our goals each week and ending up with a system that the user can utilize.
The section also shows where all the located resources, risk assessments, stakeholders, etc are.
For more information, see our GitLab repository:
https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/Project%20Plan%20Part%202.md

# Teamwork organization

Every Monday the team had a meeting where we discussed ideas, for instance: Is this idea good, why/why not? Are we able to do this? Do we have all the needed material, if not where can we get it?

At the beginning of the project the team had some meetings with the lecturers, but we were very clear about what we wanted to do, which ended up with only a few meetings with the lecturers. We have been communicating with the lecturers throughout the project.

After the usual presentation on Mondays by the lecturers, the group had a summarization of what every individual had been doing regarding the project. One student started to present the work he/she had done, and then the other group members could ask questions, or for more detail. Some of the time, we delegated one subject to two people, depending on the difficulty of the task.

The purpose of the group meetings was to get everyone informed about what is going on with the project, what they're doing and what they have been doing, and also a reminder that everyone in the group should have more or less the same knowledge. The group created some "issues" (on the GitLab project page) at the beginning of the project, but forgot about this, and didn't follow up on it properly. Later on, we realized that it simply wouldn't do, and we all sat down and created a lot of new "issues". We had a meeting where everyone in the group participated, and we delegated tasks so everyone in the group knew what to do. From then on it was smooth sailing.

# Use cases

Use case 1: Operate the nerf gun.
Actor:  Operator.

Basic flow:

1. The operator puts darts in the magazine, fully loading it, and inserts it into the nerf gun.

2. The operator presses the spin-up button and pulls the trigger to launch the darts.

3. The display will show the number of remaining darts in the magazine.

4. The buzzer will make a brief sound when halfway through the magazine, and two quick sounds when no darts are remaining in the magazine.

Alternative flow:

When there is a jam, the nerf gun will not fire the darts. The operator will open the jam door, clear the blockage and close the jam door.

Use case 2: Check statistics.

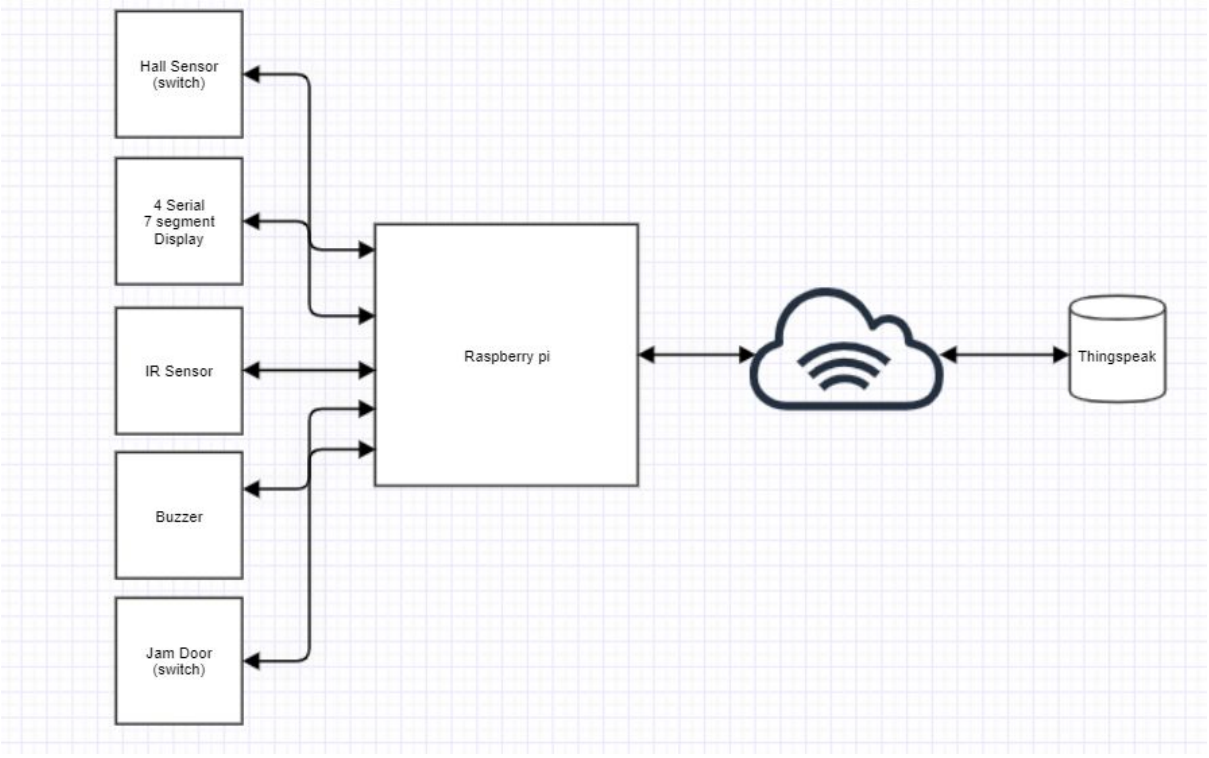
Actor: Operator.

Basic flow:

1. The operator will log on to Thingspeak.

2. The operator will check for the number of magazines used, total number of jams, and total darts launched as recorded by the sensors.

# Block diagram



In the diagram above, all the components communicate with Thingspeak through the raspberry pi, to the internet, and finally to the Thingspeak. When the user changes the magazine, the hall sensor will detect the change in the magnetic field and send a signal to the system (raspberry pi).

Whenever the signal between the IR transmitter and the IR receiver is blocked, the sensor will send a signal which will initiate the countdown. The 4 serial 7 segment display receives signals from the system, which initiates the countdown. When the magazine is half empty, power will be sent to the buzzer to make a brief tone. Once the countdown reaches 0 (empty magazine), power will be sent twice to the buzzer to make two quick tones. The jam door works as a switch here. When the jam door is opened, it triggers an event which the system will pick up, and then counts it as a jam. All the communications from the components get sent to the raspberry pi, which then returns the signal between the components and Thingspeak via the internet.

# Recreation manual

This recreation manual has been made to provide any person willing to make the nerf gun attachments the ability to do so. These modifications and attachments are designed to equip your nerf gun with sensors necessary to provide you with a shot-counter and magazine alarms. It will also be able to send data to Thingspeak, which the user can see for themselves.

This guide consists of step by step instructions on how to prepare the gun, 3D print the necessary parts, as well as the correct placement of sensors. It will also contain tips and tricks regarding the wiring and soldering.

The guide itself can be found in our GitLab repository:
https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/Recreation_manual.pdf

# Documentation of system tests

A comprehensive document detailing the challenges the team faced with each component has been compiled and put into our GitLab.

Nearly every component came with its own set of challenges, as most of them were components we have yet to work with during classes. Sometimes the hindrances was simple incompatibilities, other times it was assumptions that were proved wrong, and one time, it was as simple as just adjusting a variable resistor.

Link to the documentation can be found in our GitLab repository:
https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/Documentation/Documentation_of_tests.pdf

# User manual

In this guide, the user will be shown all the various significant parts, the operation of the nerf gun, and the safety precautions that must be adhered to when operating the nerf gun.

The user guide for this product will be slightly different from the ordinary nerf gun since extra features have been added to the original product.

The full user manual in PDF form can be found in our GitLab repository here: https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/User%20Manual.pdf

## Attachments

*The following list contains the modified attachments for the nerf gun*:

1. A display and a buzzer for darts remaining in the magazine (7-digit display and a buzzer).
2. Dart counter (infrared sensor).
3. A sensor attached to the jam door to initiate the operation (jam door).
4. Magazine.

## User guide

1. Load the magazine and insert it into the nerf gun.
2. Hold down the safety-button and pull the trigger to start firing.
3. The digital counter will initiate the countdown for the number of darts remaining.
4. The Buzzer will make a short tone when only five darts are remaining in the magazine.
5. The Buzzer will make two short tones when the magazine is finally empty.
6. Reload the nerf gun and repeat this routine
7. The system will be live on Thingspeak which means that the user can see statistics.

# Final product and casing


Three attachments are printed for the nerf-gun:

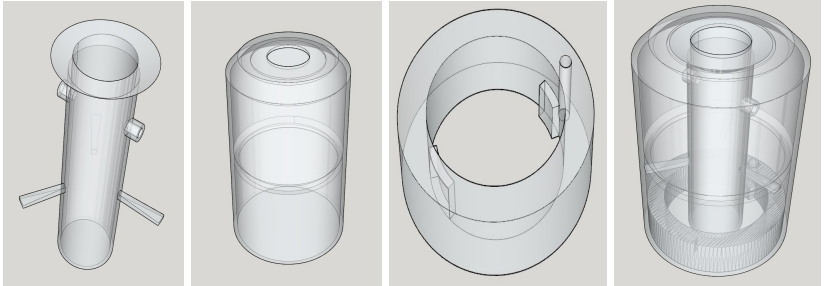
1. One is attached to the barrel.
2. One is attached to the handguard rail.
3. One is attached as a “scope”.

# Barrel attachment

The barrel attachment consists of three 3D printed parts and two electrical components.

## Printed parts

The three parts consist of an inner tube, an outer tube, and a locking mechanism.



### Electrical components

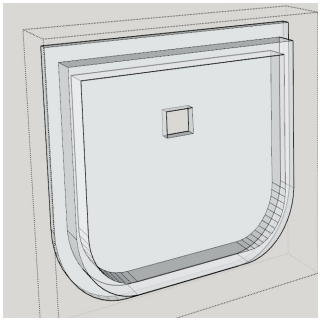
The two parts are a LIR034 3mm infrared LED and the TCRT5000 phototransistor.

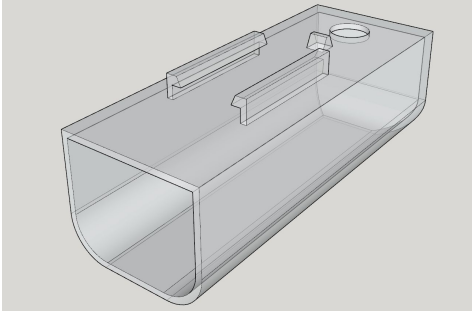
## Handguard rail attachment

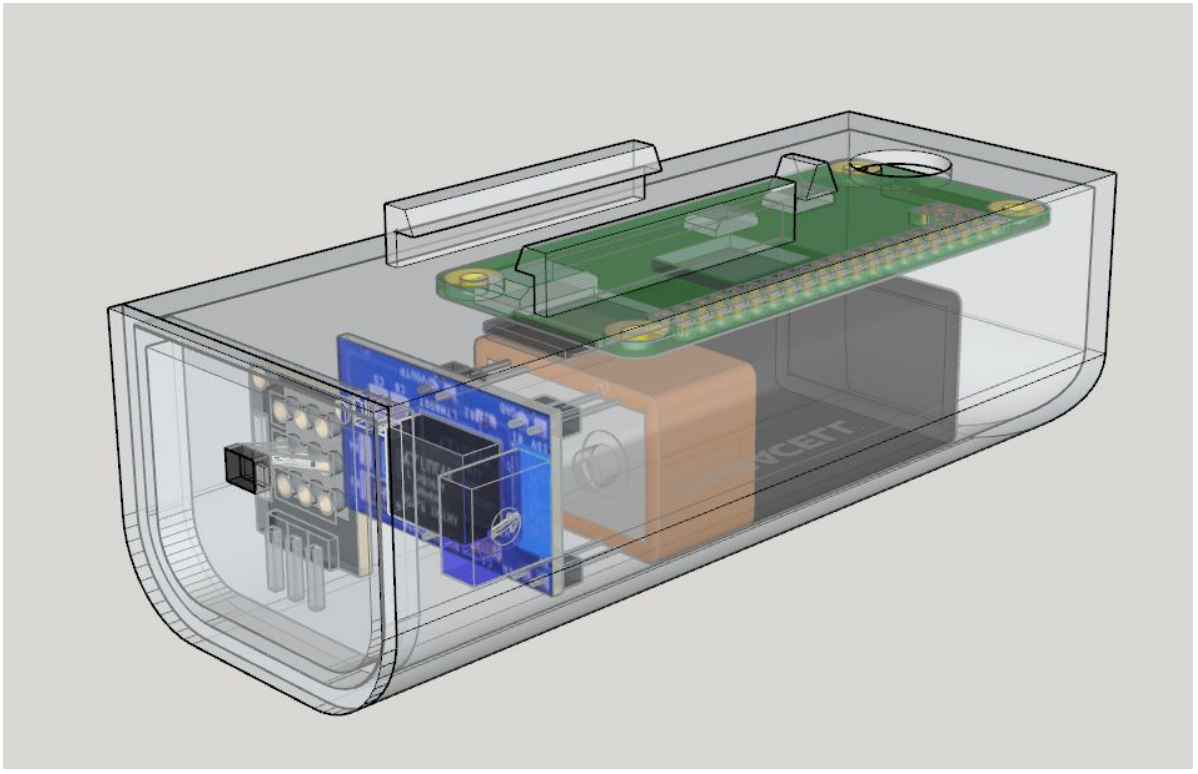
The handguard rail attachment consists of two 3D printed parts, three electrical components, and a 9V battery.

### Printed parts

A housing for components and a “plug” is printed.







### Electrical components

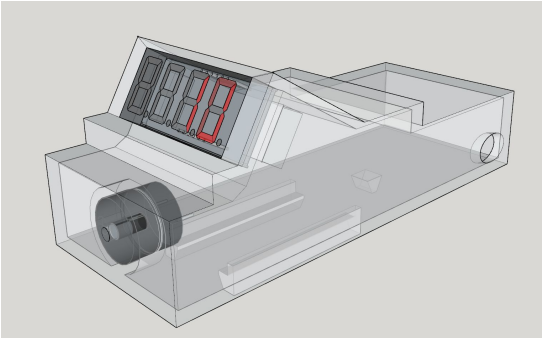
The three electrical components are a Raspberry Pi Zero W, an LTM8067Y isolated DC-to-DC converter (BOB), and a 3144 sensitive hall-effect switch (BOB).

# “Scope” attachment

The “Scope” attachment consists of one 3D printed part and two electrical components.

## Printed parts

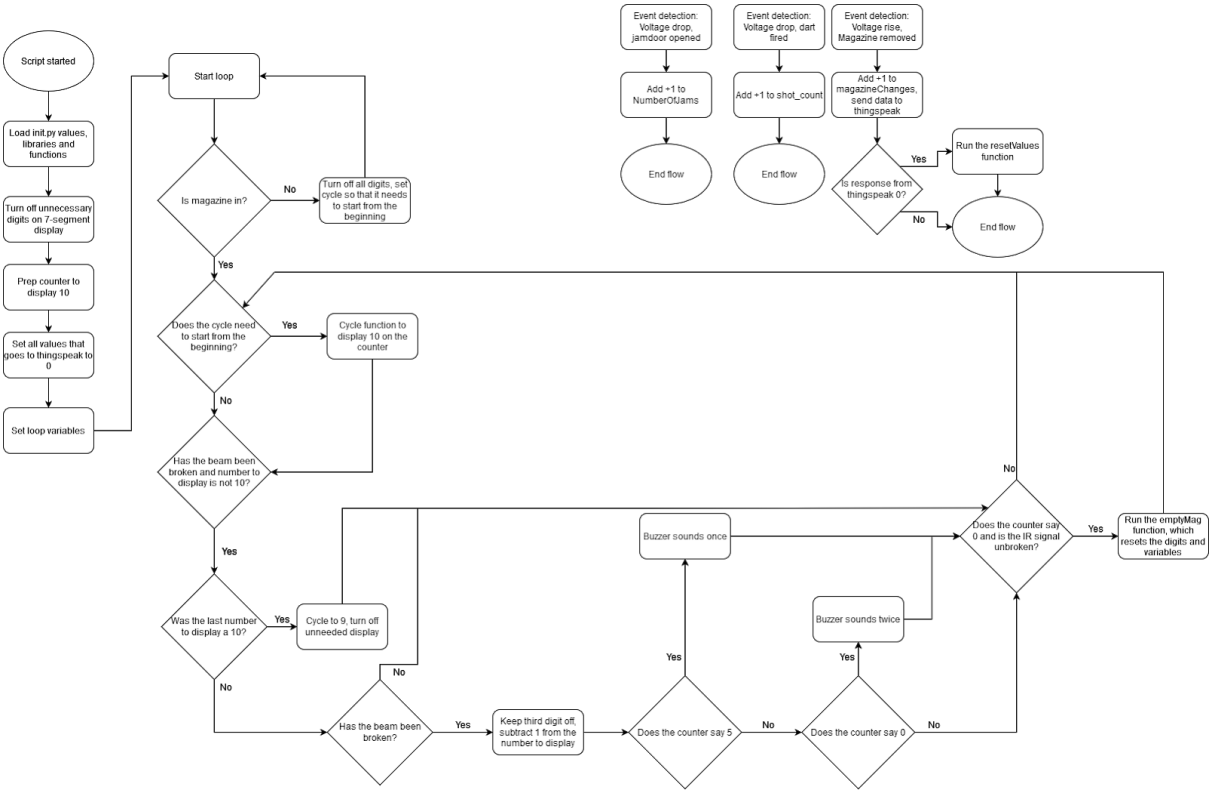
A single housing for the seven segment display and the buzzer is 3D printed.



## Electrical components

The two electrical components are a 3461BS-1 4-digit 7-segment display, and a 12mm diameter active buzzer.

# Codeflow



When the code starts, the first thing it does is that it loads the library 'init.py'. This library contains all the other libraries which are *RPi.GPIO*, *time*, *thingspeak*, *os,* and *dotenv*:

`from init import *`
This is our own library in the root folder. We choose to import all, since we are using every single thing in the script.

`import RPi.GPIO as GPIO`
For interfacing with our Raspberry Pi.

`import time`
Time is used to introduce delays into our program.

`import thingspeak`
Used to communicate and send data to Thingspeak.

`import os`
Allows us to easily get environment variables through the operating system.

`from dotenv import load_dotenv`
Used to load environment variables, which in this case is better than having a file for every key and IP.
With "*load_dotenv*", we fetch our Thingspeak API write key, channel ID, and field ID's from the .env file in the same location.

The next thing we do is that we turn off all unnecessary digits on the 7-segment display, that we will not be needing while keeping the others turned on. We do that by running the "*turnOff*" function.
The first number we need to have displayed is "10". To do this, we need to turn on each digit by itself, while changing the number between 1 and 0, and then rotating.
To actually display a digit on our 7-segment display, we need to make use of bitmapping. We use the function "*displayNumber*", which refers to our lists in init.py.

We then need to make sure all Thingspeak values are 0, so we make use of the "*resetValues*" function. That way we can make sure all values sent to Thingspeak are correct.
We also need to set all the variables we use in our loop before we run it. As soon as those variables have been set, the loop is ready to run.

The first check we make, as soon as the loop starts, is to see whether or not the magazine is in. If it is, we can turn on the display and show our "10". If the IR beam breaks while the number is still 10, it will jump out of its cycle and start counting down from 10 to 0.
When it does that, the 3rd digit will be turned off. The display will afterward continue counting down. When it reaches 5 it will activate the buzzing, making it beep once. When it reaches 0, it will beep twice quickly. The counter will stop counting when it reaches 0.

Once the magazine is pulled out, the program will reset itself, and start over when the magazine is inserted once again.

When the magazine has been taken out, all our data collected from our functions "*jamDoorOpened*", *'magazineInserted*" and "*decrementDisplayIncrementCount*" will be sent to Thingspeak.

Everything is wrapped in a loop, so we can continue as long as we have power to our Raspberry Pi.

The code itself can be found in our GitLab repository:

**main.py:**
https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/Code/Complete_Code/main.py

**init.py:**
https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/Code/Complete_Code/init.py

# Thingspeak

As part of the project, thingspeak had to be used to collect data and display it in a meaningful way. We choose to do this through the usage of summed bar graphs, which will display the daily changes of magazines, the number of darts fired and the number of times the jam door has been opened, signaling a jammed gun. The brief documentation can be found here:
https://gitlab.com/20a-itt1-project/team_b2/-/blob/master/Project_Nerf_Alpha/Documentation/Thingspeak_documentation.pdf

While the specific channel used can be found here:
https://thingspeak.com/channels/1278861

# GitLab page

https://20a-itt1-project.gitlab.io/team_b2

# Improving and expanding upon the product

After the product has been completed, the first thing to do will be to improve it. As of now, there are a lot of exposed wires and loose components, which are not super easy to work with. The full product itself requires direct modifications of the nerf gun, while the attachments are custom 3D printed, and currently only available for people with access to a 3D printer.

A huge improvement would be to condense the attachments needed while eliminating the need for wires outside the enclosures.
There are several ways to expand upon the product. Those that would be considered realistic for us for now, would be to maybe add a small OLED display on the side of the

nerf gun, displaying the information we are collecting from Thingspeak directly to the user. The Raspberry Pi Zero already has a Wi-Fi module, so it is able to receive data as well.

Another way to expand upon the product would be to add an infrared measurement system, measuring the distance between the user and its target. This distance could also be displayed on the OLED screen, or a separate counter similar to the 7-segment display already used.

Regarding data collected, the original goal was to also implement a way to see mean time between failures on Thingspeak, but this was dropped because of time constraints. There are currently 3 bar graphs displayed, but it would have been better to have simply numeric displays with accumulated data. This was however too challenging to implement, as it would need extensive knowledge of the Matlab language. In the future, an automated way to generate a channel (or maybe even a whole new website) for the data collected would definitely be an improvement, as currently, it has to be done manually.

While putting the final touches on the product and assembling it, we discovered that we didn't implement a proper on/off switch.
Looking towards the far future, the product might even branch into other "gun mod" markets, such as the airsoft community. This would however require many improvements on both size, modularity, ease of assembly (or even selling already assembled kits), and generally everything listed above. That would be the far-term goal.

# Conclusion

In the second part of the semester project for ITT1, we took a nerf gun from the product line Stryfe and modified it. We disassembled it, studied the interior and got an understanding of how it works. Then, after a long process of planning and work, the group came up with the idea of adding a shot-counter to the nerf gun.

The original goal was to be able to count the amount of darts being fired. This was later expanded upon to include amounts of magazine changes, and count how many times the jam-door for the nerf gun had been opened. All this data would then be sent to Thingspeak while displaying the remaining darts in the magazine to the user on a 7-segment display. We achieved this goal in the end, but not our secondary goal of wanting to convert and apply the product to a BB gun.

Enclosures for all the hardware and components required were designed, and a quick proof of concept was made. Due to the COVID-19 pandemic, the project proved to be harder than expected, affecting several factors such as physical project construction and organizing.
This is also the reason for only having a rough prototype working, by the date this report was made, and not a finalized sellable product. We are however going to continue to work towards completing the project.

In conclusion, the project proved to be a challenge for all of us, which in the end made overcoming it more rewarding and leaving us with a product we are proud of.