freETarget

Part 1 – Background and Theory of Operation

What is freETarget and Open Source?

freEtarget began around 2016 at the NRA sectionals in Beatrice Nebraska. In a casual conversation with one of the competitors, he mentioned in passing that he built an electronic target using microphones. Intrigued, I began looking into the subject but did not get very far before life intervened and I put it aside. Some years later I had cobbled together a circuit and some software that helped to define the scope of the work, but only went to show that there was a lot more left to do. Enter another conversation with a friend who was working on an open source project to write an electronic circuit design program. His project had gone on for many years and had resulted in one of the most popular programs used by makers for building circuits. So maybe the way to get freETarget finished was to make it open source.

Open Source began in the late 80's by software developers who saw that operating systems were becoming the property of large companies. Their solution was to write their own operating system and publish the source files for others to use. The most common example is Linux, which went from an obscure platform in the 90's to the most prevalent platform today. Examples of open source that you probably use every day include

- Android smartphones
- Linux operating system
- Libre Office business software
- Raspberry Pi computers

Contributors to freETarget are in Romania, USA, and Germany. freETarget systems are working on every continent except for Australia and Antarctica. There are users in countries that I only read about in geography class.

What are the Design Requirements?

No successful project can begin without a solid set of requirements. Without them, the project devolves into chaos and never gets finished. So with that the basic set of requirements are:

- Be capable of resolving an air rifle shot to ISSF requirements
- Be assembled in developing countries using local materials
- Be built as a Eagle Scout or 4H project
- Be accessible to users anywhere in the world

The result has been some controversial design decisions that seem at odds with our high-tech lifestyle.

The ISSF requirement means that the projectile position has to be measured better than 0.25mm. Converting that to the speed of sound means that the time measurement is less than one microsecond. Adding in an error budget means that freETarget is measuring time in 1/8 microsecond increments.

Building using local materials meant that the target holder could be made from plywood and cardboard. To accommodate the assembly tolerances, the circuit is built so that only two measurements are important.

Since it is hoped that freETarget would become an Eagle Scout project, the technical expertise required to build it was pegged at a middle school level. This meant that the connection between the target and the firing point would have to be easy to install. While it is low tech a long USB cable solves a lot of problems.

Everybody seems to have an opinion about what constitutes the best computing platform. Worst is that freETarget needs two computers; one in the target holder to work out the projectile position, and another at the firing point to display the shot information. Choosing a PC for the firing point display was an easy choice, PCs are available worldwide, and they all work the same way. The selection of an Arduino was more contentious, but in the end Arduinos are available at a low cost and the development tools are available on the web for free.

So in the end, freETarget became:

- Based on a PC
- With software written in C using Visual Studio
- Uses an Arduino Mega 2560
- Supported by a purpose built signal acquisition shield
- Available for download on GitHub (ten-point-nine/freetarget)

Theory of Operation

The basic problem that freETarget tries to solve is to triangulate a position based on the speed of sound and the time between sensors.

freETarget uses four microphones located at the cardinal points North through West as shown in Figure 1. While moving through the air the pellet creates a disturbance that radiates out circularly from the pellet. The disturbance is picked up first by the closest sensor and last by the furthest one. As each sensor detects the disturbance it starts a timer and about a millisecond later all of the timers are stopped and read out. Subtracting each of the times from the largest one gives a measure of how close the sensor was to the pellet.

Unfortunately the times do not give you any idea how far the pellet was from the first sensor. Fortunately the time offset for each of the sensors is the same, and this is the location of the pellet relative to the closest sensor. The shot location is found by taking a guess about the time offset and with trigonometry finding the four points that correspond to that guess. Figure 2 illustrates the first guess and four points.

The four points are averaged and provides an improved estimate of the shot location. Repeating the calculations again gives a better guess at the shot location. This is done until the difference between the estimates becomes small, at which time the shot location is now know.

The Arduino software takes care of reading the counters and computing the shot position. It then outputs the shot information in a readable form to the PC which then presents the shot on the screen as a target and statistics.

Circuit Design

Arduinos are low cost computers that are popular with makers. They provide a sophisticated platform with accessible development tools for about $10 retail. The Arduino MEGA 2560 has enough memory and peripherals to make building a target interface easy.

The freETarget circuit is a purpose built Arduino shield that has the following circuitry.

- Power supply for the sensors
- Projectile identification
- Timing circuit
- Target illumination circuit
- Accessory connector

For this article we'll only discuss the projectile identification and timing circuits.

The block diagram of the target circuit is shown in Figure 4.

The projectile sensors are four microphones located at the cardinal points (Figure 2). As the projectile passes the microphone, it creates a short 'pop' that is converted to a rising electrical signal. This signal is sampled by a comparator, and when it exceeds a certain value triggers a start signal. The start signal gates an 8MHz counter that runs until stopped by software. The timers are then read out into the software to be converted into a location.

Part 2 – Building freETarget

The goal of freETarget was to come up with a kit that could be sourced from local materials and assembled by an Eagle Scout. The acquisition circuit can be copied from GitHub or purchased complete. The remainder is downloaded and built at home.

Building a freETarget system is done in four steps:

- Building the target frame
- Installing the sensor circuit
- Loading and setting up the PC program

BUILDING THE TARGET FRAME

The target frame holds the target and sensors in the correct position to record the shot. The assembly can be made of anything local, such as

- Plywood
- Plastic
- Metal
- 3D Printing
- Cardboard

A side view of the target assembly is shown in Figure 1.

Item	Description	Notes
Front Face	Face plate used to protect the circuitry and sensors	Should be made from a material that will withstand a pellet strike
LED Illumination	LED lightning strip attached to the front face to light the target	Can be obtained from a local hardware store
Face Sensor	Duplicate sensor mounted to the front face to detect a shot to the front face	Operation dependent on the material used for the front face.
Sensor	Four microphones located around the target to detect the shot	Pay careful attention to the locations stenciled on the circuit board. Sensors may be located anywhere between the front face and pellet

		trap.
Flat Cable	Routes from the Arduino board to the sensors	End marked with an A is the Arduino end of the cable
High Pass Filter	Small board that installs between the flat cable and Arduino	Used to attenuate report from gun
Sensor Block	Mounting block to hold sensor parallel to direction of shot	May be any non-conductive material.
Sensor Support	Back frame to hold the sensors in the correct location and orientation	
Target	Target	May be located anywhere between the front face and pellet trap.
Pellet Trap	Pellet Trap	Used to collect pellets after shooting. May be omitted if shooting on a range with a berm.
USB Cable	Fifteen meter USB cable	May be purchased locally or on line

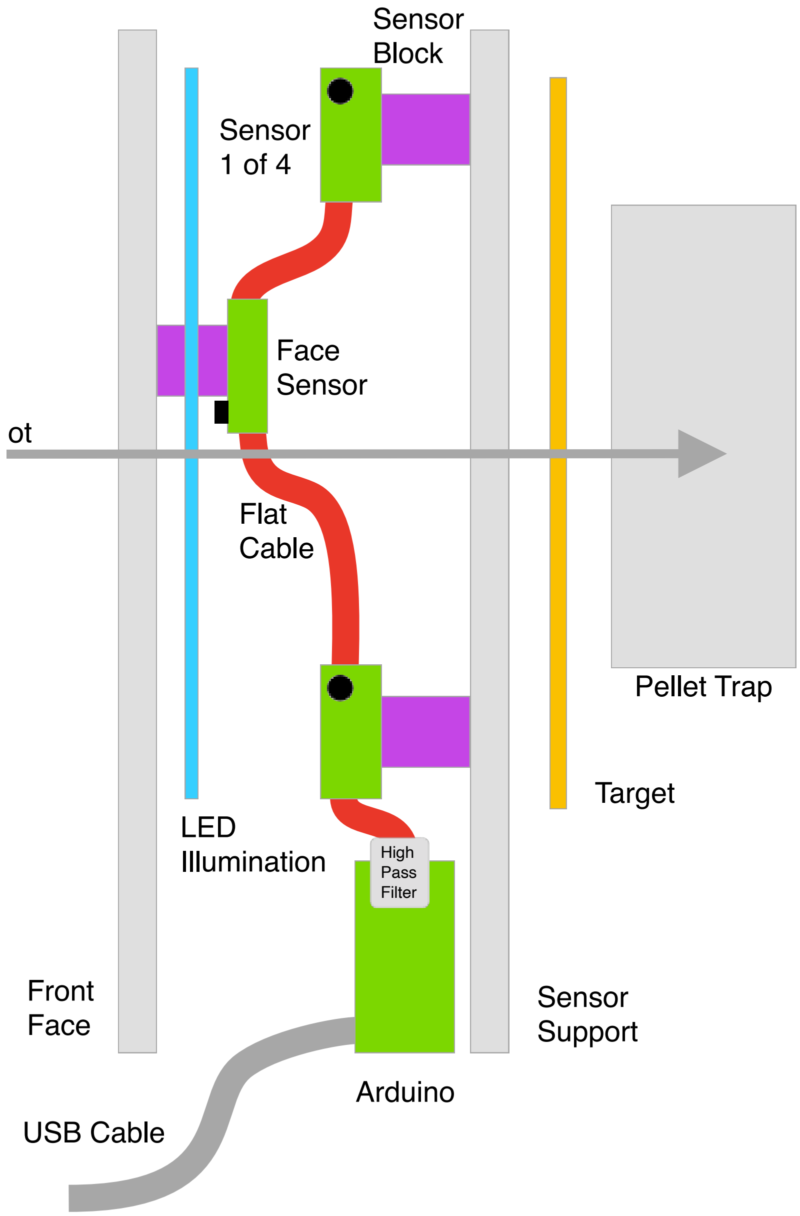


Figure 1: freETarget Layout

The only thing that is important is that the sensors be mounted at the corners of the target at a distance of 230 mm sensor-to-sensor. Figure 2 illustrates the geometry of the sensors as viewed from the firing point

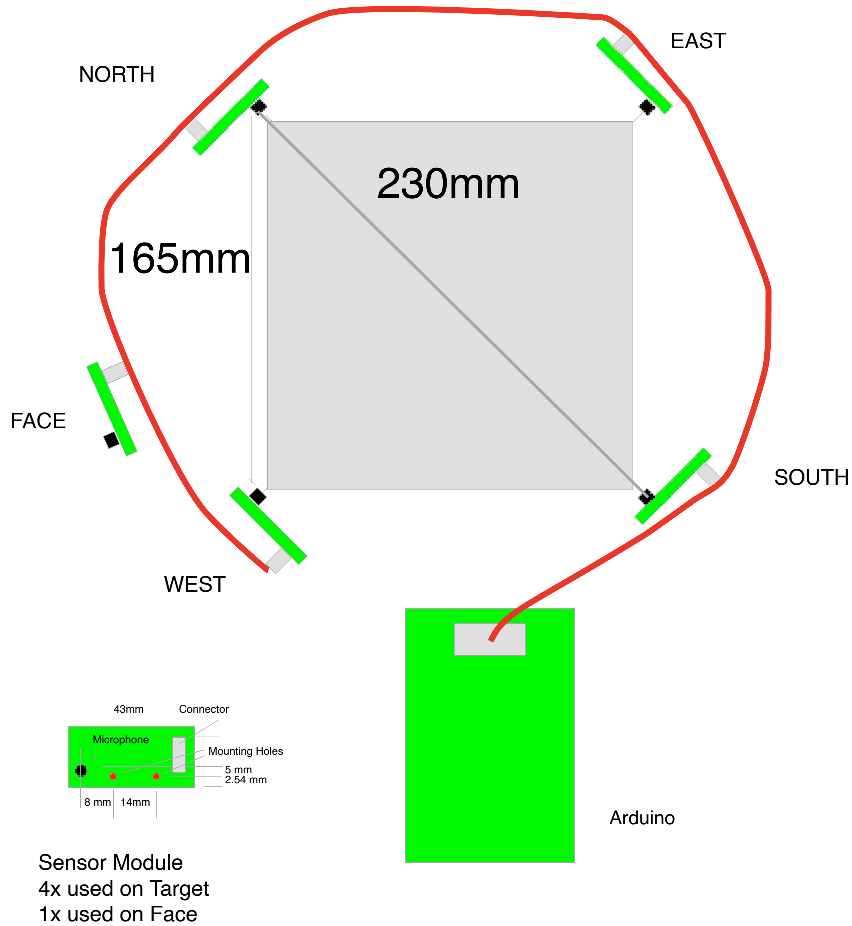


Figure 2: Sensor Geometry

The easiest construction method is to use plywood sheet. Figure 3 is an illustration of a target holder.   Figure 4 shows a sample sensor block

Begin by cutting a rectangular frame and square target hole. The frame can be as large as you like, and the target hole has to be big enough to hold the target in place.

The sensor mounting blocks are rectangular blocks of plastic 12mm x 25 mm x 27mm. Drill two holes in sensor block to mount the sensor boards then attach the sensor blocks to the target frame using epoxy.

Locate the bullet trap behind the target holder.

INSTALLING THE SENSORS

The next step is to install the sensors.
Note that the sensors are marked

- NORTH
- EAST
- SOUTH
- WEST
- FACE (more about that later)

Install each of the pellet sensors as shown in Figure 5 Sensor Installation

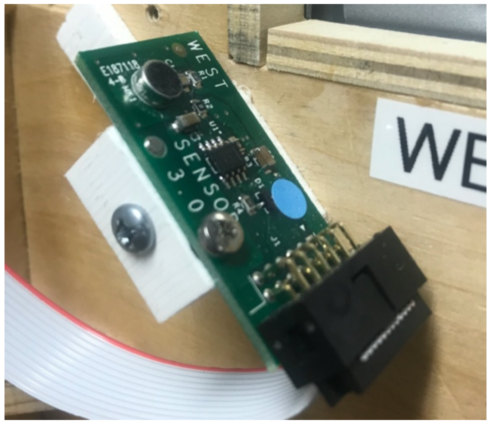


Figure 5: Sensor Installation

Install the Arduino at the bottom of the target holder and install the flat cable around all of the sensors as shown in Figure 6.

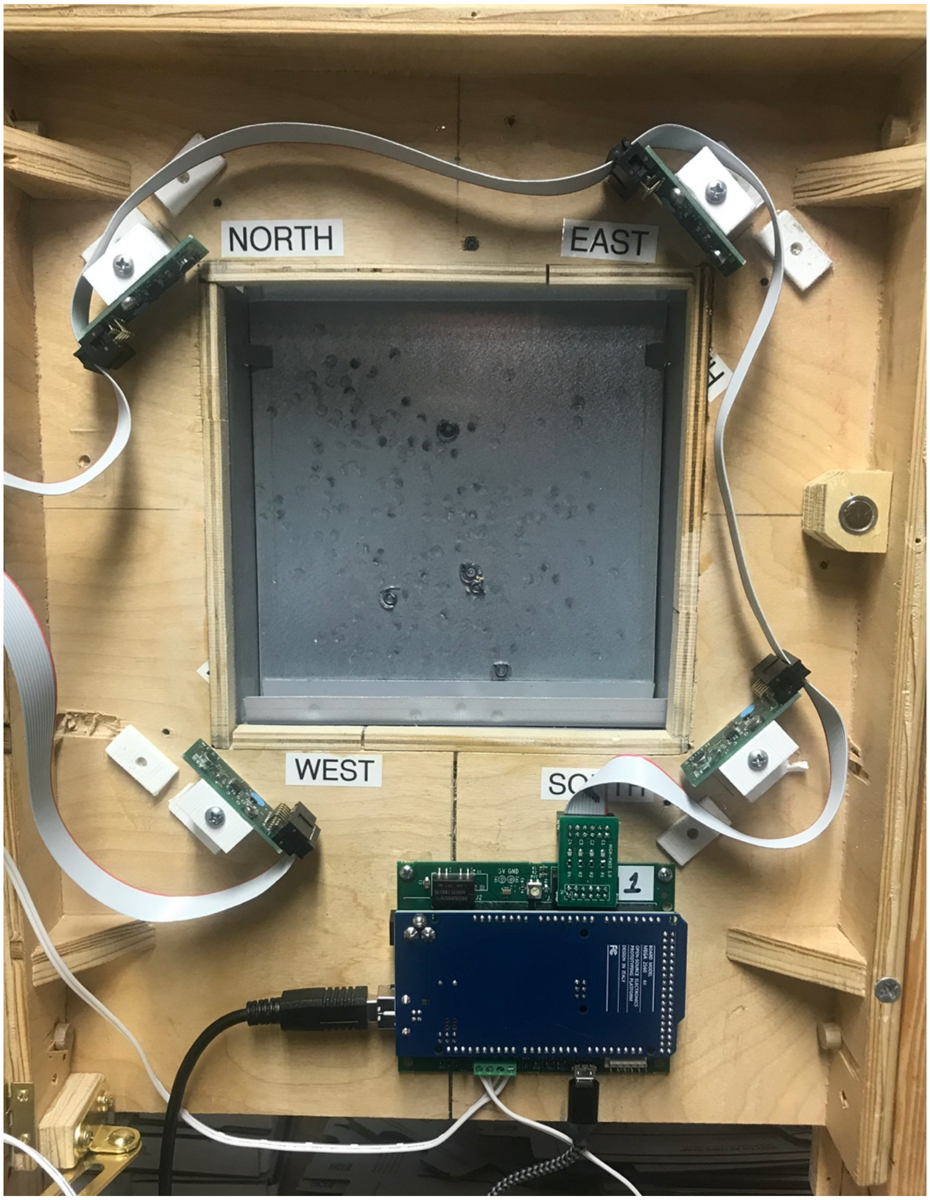


Figure 6: Flat Cable Installation. Note Location of High Pass Filter

INSTALLING THE FACE SENSOR

The face sensor detects the sound of the pellet striking the case. The face sensor works best on a metal case, and not so well on plastic or wooden structures. Depending on the material used, the face sensor may not work at all.
Attach the face sensor facing the door and with a solid material between the door and the microphone. Figure 7 illustrates the attachment.

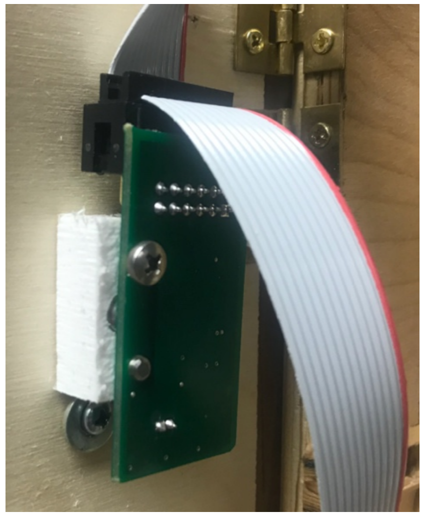


Figure 7: Face Sensor Installation

DOWNLOADING AND INSTALLING PC SOFTWARE

The PC software is available for free at https://free-e-target.com/downloads/

Look for the PC Software section and download the software (Figure 8)

PC Software

The source files are available on the Github, and you can build your own using the VisualStudio hobby edition.

The most recent version can be downloaded here:

freetarget-1.13.0_2 Download

Once you download it, unzip the files and follow the instructions.

Figure 8: Download Software

STARTING UP

Launch the PC program and look for the setup icon (GEAR WHEEL) in the upper right corner (See Figure 9)

Figure 9: Setup Icon Location

Enter all of the setup information needed in Figures 9, 10, 11, and 12

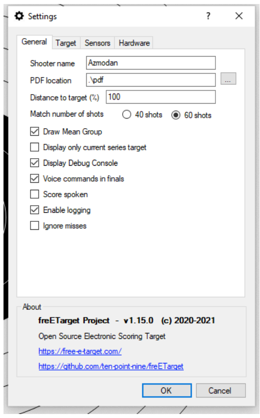


Figure 10: General Settings

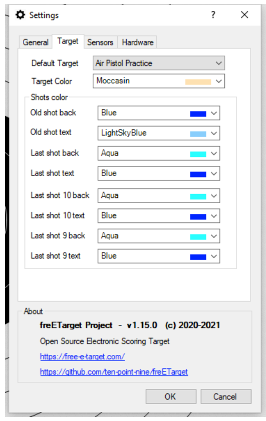


Figure 11: Target Settings

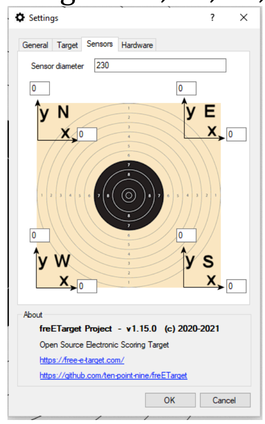


Figure 12: Sensor Adjustment

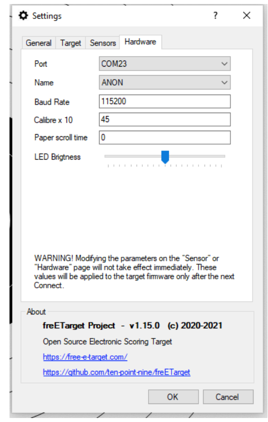
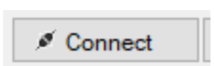


Figure 13: Hardware Interface

Figure 10: This allows you to enter the shooter name and how information will be stored.
Figure 11: Choose the target you will be shooting against and the colours you will be using.
Figure 12: Fine tune the sensor position to adjust for assembly errors
Figure 13: Interface to the target hardware.

Press  to begin a session.

Refer to the Commissioning Instructions from the web site. This will give you a quick summary of how the system is working