Project Abstract:

In match shooting competitions, targets placed 500+ yards from the marksman are very difficult to see and the location of a bullet strike at that distance is impossible to see without the aid of a spotting scope. At distances greater than 500 yards, a ¼ inch bullet hole is impossible to see even with the best spotting scopes. Our system will allow the shooter to see his bullet impact within seconds of the impact taking place – which, itself, could be over a second after the marksman has fired – at distances up to 1000 yards. We expect to achieve this by having a remote unit near the target that will record impact data and then transmit the data to a unit near the marksman. Additionally, the marksman needs the ability to aim the camera at the target end, requiring two-way communication over the distance of 1000 yards. Technical challenges include keeping the target unit power as low as possible since we don’t anticipate power to be readily available near the targets, how to achieve two-way communication with the required bandwidth over the specified distance, interfacing the remote camera and extracting impact information from the images.
Table of Contents
Features ........................................................................................................................................ 3
Analysis of Competitive Products................................................................................................ 3
Concept/Technology Selection ........................................................................................................ 3
Project Architecture ..................................................................................................................... 4
Software Flow Chart .................................................................................................................... 6
Division of Labor ............................................................................................................................ 7
Bill of Materials .............................................................................................................................. 7
Gantt Chart ................................................................................................................................... 8
References ..................................................................................................................................... 9

Table of Figures
Project Architecture ......................................................................................................................... 4
Software Flow Chart .................................................................................................................... 6
Division of Labor ............................................................................................................................ 7
Bill of Materials .............................................................................................................................. 7
Gantt Chart ................................................................................................................................... 8
Features:
The E-Scope system will be able to:
- Take an image of a target with enough resolution to see impact data.
- Transmit the image to a receiving station up to 1000 yards away.
- Be aimed remotely by the shooter.
- Low level efficient implementation on camera unit.

Analysis of Competitive Products:
Even though there are similar products, they do not target this niche market and are both cost and licensing prohibitive.
- Black Widow AV
  - Need an amateur radio license to use this product.

Concept/Technology Selection:
- Logic Level
  - Leads to a streamlined final product.
  - FPGA prototyping has little to no overhead cost.
  - Familiar to team members.
- Camera and Development Board
  - The camera has a high enough resolution.
  - Documentation was comprehensive enough for a beginner.
- XBee
  - A very simple serial interface with an acceptable range.
- C#
  - Simplest way to develop a Windows based application.
  - Familiar to team members.
- Servo
  - Reliable, well-documented, in-stock servo.
- Other technologies
  - All other cameras were not documented as well.
  - Propeller processor was geared more for signal processing and heavy parallel computation which was not the goal of this project.
  - C++ is not the ideal language for Windows based application development.
Project Architecture:

Powering on the camera module sends a reset signal to the XBee to clear its output buffer. The FPGA then waits for a serial signal in the form of a single byte. This byte commands the FPGA to start one of three possible branches:

1. **Up** – Moves the servo 1/32 of its travel range, causing the camera’s field of view to rise. Using the FrameValid and LineValid output signals from the digital camera, the pixel timing is set and a frame is output from the camera to the RAM on the DE1, one out of every 16 vertical and horizontal pixels are saved. The RAM is then sent an address at a time to the serial UART to the XBEE and transmitted wirelessly to the laptop.
2. **Down** – Identical to “Up” with the exception the servo travels in the opposite direction, causing the camera to tilt down.
3. **Get Image** – In this branch, the image is downsampling by 8, resulting in a larger image transmitted in the same manner as the “Up” and “Down” commands.
When the laptop’s GUI is started, the user has the option to click one of three buttons:

1. **Up** – Sends a single byte via the XBees to the DE1 where the byte is interpreted and acted on. The laptop then receives a byte array with pixel intensity values. This array is then rendered in a bitmap format and displayed in the image section of the GUI. The receive time for this image is about 23 seconds.

2. **Down** – Same as Up.

3. **Get Image** – Same as “Up” and “Down” except the byte array is four times larger and the corresponding image has a higher resolution. The receive time for this image is about 1 ½ minutes.
Software Flow Chart

Start

Take low-res picture and display it.

Default State

Command Tree

Reposition

U or D

Move U Move D

Shot Fired

Take Large Image and display it.
Division of Labor:

<table>
<thead>
<tr>
<th>Item</th>
<th>Charlie</th>
<th>Scott</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research/project proposal</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Analog Circuit Design</td>
<td>10%</td>
<td>90%</td>
</tr>
<tr>
<td>Digital Circuit Design</td>
<td>90%</td>
<td>10%</td>
</tr>
<tr>
<td>Software/Code</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Circuit Simulation</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Purchasing Parts</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>User Interface</td>
<td>25%</td>
<td>75%</td>
</tr>
<tr>
<td>Breadboarding</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>PCB</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Assembly of Prototype</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Final Report</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Demo</td>
<td>50%</td>
<td>50%</td>
</tr>
</tbody>
</table>

Table 1: Division of Labor – We were both involved in almost every aspect of this project.

Bill of Materials:

<table>
<thead>
<tr>
<th>Component</th>
<th>Qty</th>
<th>Unit Cost</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCD Camera</td>
<td>1</td>
<td>$205.00</td>
<td>$205.00</td>
</tr>
<tr>
<td>Xbee Chips</td>
<td>2</td>
<td>$32.00</td>
<td>$64.00</td>
</tr>
<tr>
<td>USB Kit for Xbee</td>
<td>1</td>
<td>$25.00</td>
<td>$25.00</td>
</tr>
<tr>
<td>Battery</td>
<td>2</td>
<td>$4.00</td>
<td>$8.00</td>
</tr>
<tr>
<td>Misc Electronic Comp</td>
<td>1</td>
<td>$20.00</td>
<td>$20.00</td>
</tr>
</tbody>
</table>

| TOTAL                      |      |           | $417.00 |

Table 2: Bill of Materials
Table 2: Gantt Chart

Table 2 is the actual Gantt Chart. We clearly underestimated the time it would take for the code, user interface, and breadboarding.
References:

