I.F.F. (Identification Friend or Foe) System

By

Eric Meyers (emeyer7)

Noah Prince (nprince2)

ECE 445 Final Report - Spring 2016

TA: Braedon Salz

May 4th, 2016

Project No. 11

Abstract

This piece of equipment will reliably determine the status of a target. FUCK I DONT KNOW. LIKE HOLY FUCKING SHIZ YO This project is a reliable method to determining the status of friendly or enemy soldiers during combat eee reduce the number of friendly fire or misfire accidents between soldiers on foot during combat. A “friend or foe” detection

Contents

1 Introduction

The purpose of this project is to create a system that quickly and accurately identifies friendly targets among military personnel on foot. Similar systems exist for aircraft, however not many exist for infantry.

The idea is to develop a two-way communication system so that when a soldier aims their weapon in the direction of a friendly target, they will receive notification through an LED that the target is friendly and not an enemy. Throughout this document the infantry unit with the weapon will be referred to as the "friendly interrogator" and the target will be referred to as the "friendly target".

1.1 Objectives

1.1.1 Goals and Benefits

- Reduce the number of friendly fire & misfire accidents during combat.
- Notify friendly personnel of friendly target when aiming in their direction.
- Other applications include paintball, airsoft arcade laser tag, and various recreation sports.

1.1.2 Functions and Features

- Laser transmitter on friendly interrogator to send unique I.D.
- Photodiodes on friendly target detect unique I.D. and verify its signal.
- R.F. transmitter on friendly target to send acknowledgement back to interrogator.
- R.F. receiver on friendly interrogator to verify that the target is friendly.
- LED to indicate friendly or enemy on interrogator unit with system response of less than 190 ms (human reaction time[3])

These functions and features are summarized in the system block diagram shown in Figure 1.

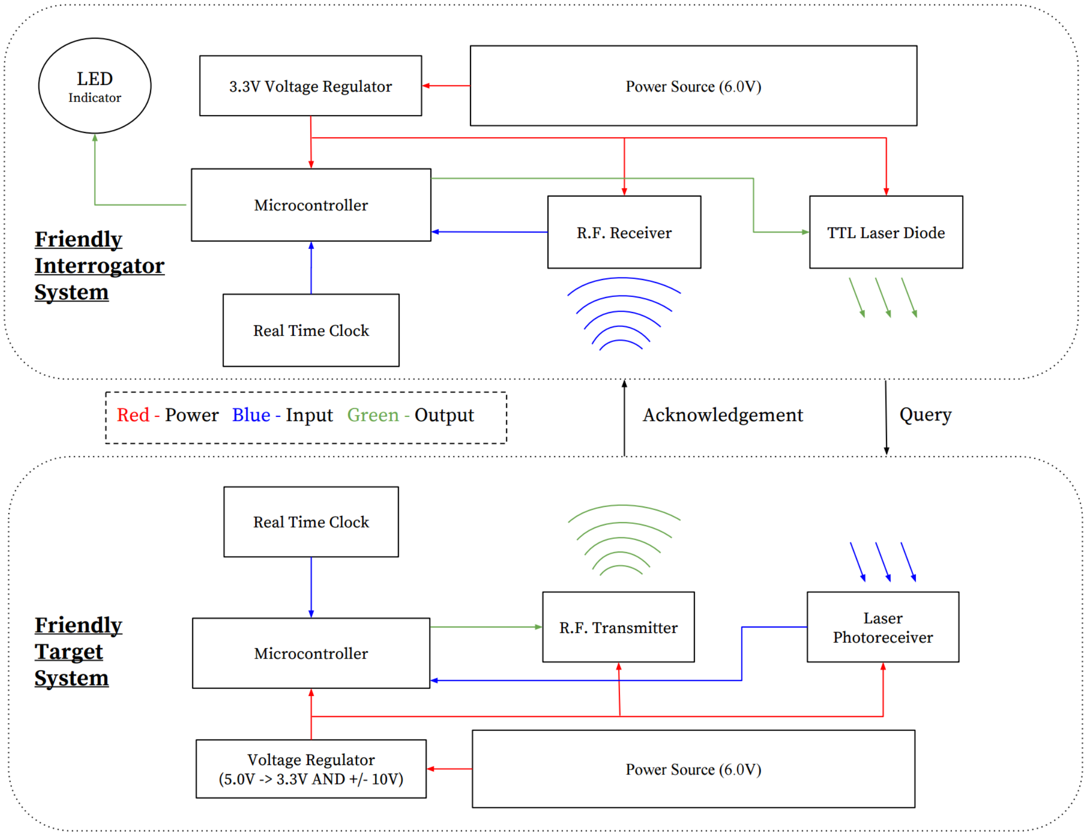

Figure 1: System Block Diagram

The two-way communication system on both units is further divided into two one-way communication channels. The laser transmitter on board the friendly interrogator will send a signal to the photoreceivers on the friendly target. The R.F. transmitter on board the friendly target will then send acknowledgement back to the friendly interrogator.

An important aspect of this project is encryption and ensuring an enemy cannot pose as friendly to the interrogator. This will be addressed in two ways. Both systems will contain a locally synced clock so that ...

1.2 System Level Requirements

Requirements are imposed on both the R.F. subsystem and the laser subsystem to accurately receive packets. From a system perspective, the requirements are as follows:

1. R.F. Transmitter/Receiver - Must be able to both transmit and receive at least 90% of 8-bit packets sent over a distance of 5 meters with a carrier frequency of 315 MHz 50 MHz.
2. Laser Transmitter/Receiver - At least 90% of transmitted laser packets must be received by the photoreciever at 5 m.

Requirements are also imposed on the system speed. These are

1. Speed of System - A friendly target at 5 m should be marked friendly within 190 milliseconds.

2 Design

2.1 Design Procedure

2.1.1 Friendly Interrogator

Voltage/Power Regulation

TODO

Laser Transmitter

For safety reasons, the maximum allowable power for the laser diode is $5mW$; which registers as a Class IIIa laser. The laser diode must also fall in the visible range, so that it will trigger a person's blinking reflex before eye damage occurs. Specifically, the team will use a red ($635nm$) laser. See Section **??** for more on safety of the laser.

The team used a 5 mW 635 nm laser diode to transmit the unique I.D. (as specified by the 8-pin DIP switch) to the friendly target. The laser diode, sourced by a transistor, allows for pulsing of a unique identification number at $5kHz$. The limiting factor for the $5kHz$ requirement was the processing speed of the $MSP430$ microcontroller; between every sampling of the photoreceiver, a significant amount of processing must occur.

R.F. Receiver

TODO

2.1.2 Friendly Target

Voltage/Power Regulation

See 2.1.1

Laser Photoreceiver

A photodiode was chosen such that a 5kHz signal could be processed and boosted to register a value between 0 and 3.3V at a maximum distance of $30m$ from the laser source. This couples the photoreceiver requirement with the intensity of the laser diode, which is capped at $5mW$. A detailed analysis of the choice of photodiode is in **??**

R.F. Transmitter

The requirement driving both the R.F. transmitter and receiver was the ability to broadcast and receive packets (as a pair) at the maximum distance of the project ($30m$). TODO: Describe some of the governing equations

2.1.3 System

Microcontroller

The microcontroller must have enough ports to service both the R.F. boards and laser/photoreceiver inputs, have enough speed to sample and decide on a packet value received at $5kHz$, and have the ability to count seconds. A vast majority of microcontrollers fit the requirements for this design, as most come with several ports, fast processors, and a built in timer.

The design decision, then, came down to ease of use and available documentation. To this end, the team chose to use the $MSP430$ series of microcontroller. Previously, the team had tried to use $MSP432s$, but these units stopped functioning mid-design.

Other microcontroller options, including PIC and arduino were considered. Arduino was considered overkill and low difficulty for the project, PIC more difficult to work with due to less documented proprietary systems.

FROM TEMPLATE: Discuss your design decisions for each block at the most general level: What alternative approaches tothe design are possible, which was chosen, and why is it desirable? Introduce the major design equations or other design tools used; show the general form of the circuits and describe their functions.

2.2 Design Details

2.2.1 Friendly Interrogator

Circuit Schematics

TODO

PCB

TODO

2.2.2 Friendly Target

Circuit Schematics

TODO

PCB

TODO

Software

Incoming transmissions are broken into two parts, where a 1 corresponds to a high at the transmission source, and a 0 a low:

1. **Preamble** - 10101010
2. **Packet** - An 8 digit long value representing a numerical id

Values from the photoreceiver are analog values ranging from $0 - 3.3V$. Because the ambient light in the room can the photoreceiver to report $0 - 2V$ when receiving no light from the transmitter, an algorithm was created to decide binary values of a transmission based only on the *difference* between the current analog value and the analog value $200\mu s$ before.

Algorithm 1 GetCurrentBinaryValue

```
Input : currentAnalogValue, lastAnalogValue, lastBinaryValue, THRESHOLD

dif ← (currentAnalogValue − lastAnalogValue)

//Was a 0, now a high indicates a 1
if dif > THRESHOLD then
    return 1

//Was a 1, now a low indicates a 0
else if dif < −THRESHOLD then
    return 0

//No change from last value
else
    return lastBinaryValue
```

Binary values are stored in an array of length 8, since both packets and preambles are 8 bits. Because a preamble can come at any moment, an algorithm was designed both to store binary values in the array, and to always check for a preamble packet

Algorithm 2 ReceivePreamble

```
photoBinary ← new Array(8)
lastBinaryValue ← 0
lastAnalogValue ← 0
currentIndex ← 0

every 200μs do
    // Store current value
    currentAnalogValue ← analogReadPhotoreceiver
    currentBinaryValue ← GetCurrentBinaryValue(currentAnalogValue,
                                               lastAnalogValue,
                                               lastBinaryValue)
    photoBinary[currentIndex] ← currentBinaryValue
    currentIndex = (currentIndex + 1) % 8

    // Check if last 8 received values make 10101010
    lastEightValues = concat(photoBinary[currentIndex...8], photoBinary[0..currentIndex])
    if lastEightValues == 10101010 then
        ReceivePacket

    lastBinaryValue ← currentBinaryValue
    lastAnalogValue ← currentAnalogValue
end
```

Once a preamble has been received, a simple algorithm is used to capture a packet

Algorithm 3 ReceivePacket

```
packet ← new Array(8)
lastBinaryValue ← 0
lastAnalogValue ← 0
currentIndex ← 0

8 times, every 200μs do
    // Store current value
    currentAnalogValue ← analogReadPhotoreceiver
    currentBinaryValue ← GetCurrentBinaryValue(currentAnalogValue,
                                               lastAnalogValue,
                                               lastBinaryValue)
    packet[currentIndex] ← currentBinaryValue
    currentIndex++


    lastBinaryValue ← currentBinaryValue
    lastAnalogValue ← currentAnalogValue
end

return packet
```

2.2.3 System

FROM TEMPLATE: Present the detailed design, with diagrams and component values. Show how the design equations were applied. Give equations and diagrams with specific design values and data. Place large data tables in an appendix.Circuit diagrams that are too large to be readable on a single page should be broken into pieces for presentation.The full diagram may be included in an appendix.Use photographs only as necessary and treat them, along with all other graphics except tables, as figures.

3 Verification

INCLUDE R&V TABLE IN APPENDIX IN BACK
TODO

3.1 Counting Packets

The following algorithm was used to count the number of missed packets from the photoreceiver

Algorithm 4 CountMissedPackets

Input: $expectedPacketValue$

$missedPackets \leftarrow 0$
ReceivePreambleAndPacket // Get first packet to ensure transmission started
100.**times do**
    ReceivePreambleOnce // Receive next 8 bits as preamble, ignore if not a correct preamble
    if ReceivePacket $\neq$ $expectedPacketValue$ **then**
        $missedPackets{+}{+}$
end

FROM TEMPLATE: Discuss the Requirement and Verification Table from your design review. Including the table in an appendix will help avoid lengthy and tedious narrative description in the main text, which may not be of immediate interest to your imagined audience of managers. Do not discuss lowlevel requirements unless they failed to verify, or you found that they were critical in some unexpected way, or you need to makes changesfor instance, to the tolerances or acceptable ranges of quantitative results. It is important to hit the main points and explain any requirement that is not verified, but keep the discussion concise and refer interested readers to the appendix for details. Note that the design procedure, design details, and design verification can be organized in different ways. The Word template provided by the ECE 445 staff puts the first two in one chapter and the second in another; however, a separate chapter for each is also common, with chapter sections reiterating the main project components. If you do the latter, avoid unnecessary repetition of component descriptions. Another option, though rarely used, is to organize the report according to components or blocks, with each chapter describing the design procedure, details, and verification for a single component or block.

4 Costs

TODO FROM TEMPLATE: Labor cost estimates should use the following formula for each partner: ideal salary (hourly rate) actual hours spent 2.5 include estimates for electronics and machine shop hours, as applicable. For parts, use real values when you know them; make realistic estimates otherwise. List both the retail cost and what you or the department paid (in this case you may list labowned pieces as free).If the project might be commercially viable, estimate the cost of massproduction by listing bulkpurchase costs. Make sure any tables are numbered appropriately, given titles, and cited directly in the text.

5 Conclusion

TODO FROM TEMPLATE: Bring together, concisely, the conclusions to be drawn. It may be appropriate, depending on the nature of the project, to begin or end with a twoor threesentence executive summary. The reader needs to be convinced that the design will work. Summarize your accomplishments. If uncertainties remain, they should be pointed out, and alternatives, such as modifying performance specifications, should be spelled out to deal with foreseeable outcomes. Usewords, not equations or diagrams. Devote a section to ethical considerations with reference to the IEEE Code of Ethics and any other applicable code (e.g., the AMA Code of Medical Ethics for certain bioengineering projects).

6 References

TODO FROM TEMPLATE: Follow the IEEE reference styles provided in this document for various kinds of sources. If you need to cite something for which there is no example, simply use common sense and providein a neat and orderly manner emulating the IEEE reference stylethe information necessary for another researcher to find that source. References [1][3] are examples of a manual, datasheet, and web page, respectively. References [4][7] are more standard, scholarly sources: a book, chapter in an edited book, journal article, and conference proceedings. Reference [8] is a technical report, and reference [9] is class notes. Cite all references consecutively in the text, as is done here. (ECE Editorial Services provides a more detailed description of IEEE reference style on its wiki: http://go.illinois.edu/ecethesis .)

References

[1] Lieutenant Colonel Charles R. Shrader, *Amicicide: The Problem of Friendly Fire in Modern War*, 1982.

[2] W. B. Garrison, *Friendly Fire in the Civil War: More than 100 True Stories of Comrade Killing Comrade*. Rutledge Hill Press, 1999.

[3] G. T. Taoka, "Brake Reaction Times of Unalerted Drivers," *ITE Journal 59 (3): 1921*, March 1989.

Appendix