Foxhunt Beacon and Receiver
Douglas Spry
Eastern Michigan University
7/1/2013

1 Introduction

Eastern Michigan University's High Altitude Ballooning Program has demonstrated difficulties with retrieving balloon payloads after the flight. This project sought to create a light, independent, and long-lived payload locating system. This has been accomplished using a binary radio system which allows for a long range with a low current draw.

The RFM22b radios used in this project use a duplex communication scheme over the 433MHz band. This radio is rated for a range of around two miles with a typical passive antenna. These are controlled by Arduino microcontrollers.

The transmitter is an extremely simple design that only requires the two previously mentioned devices and a nine-volt battery. The transmitter simply requires the end-user to upload a sketch to the Arduino and then put a battery in and it's ready to run for up to 24 hours.

In order to display signal strength the receiver uses a serial LCD screen in conjunction with the other components. This complicates the circuit as the combined current draw of the screen and radio is too much for the current supply on the 3.3V line of the Arduino. To get around this we simply need to add in a 3.3V regulator to provide an alternate power supply. The receiver also targets a specific transmitter and allows mutliple teams to work in the same region without confusing each other.

Lastly the receiver should be used with a yagi antenna which requires specific geometry but can be built in a number of ways. I will discuss the geometry and how to build a dipolar active antenna element.

2 Components

- 2x RFM22b
- 2x Arduino Nano
- 3.3V Serial LCD
- Serial level converter

3 Transmitter

Connect the radio and Arduino as shown in Appendix A. Open the sketch in Appendix C and change the variable callsign to your callsign. Connect the Arduino to USB and upload the sketch. Connect the battery and you're done.

4 Receiver

Connect the radio, regulator, and the LCD as shown in Appendix B, but leave the serial connection for the LCD unconnected. Open the sketch from Appendix D and change the variable target to your callsign and then upload it to the Arduino. Connect the serial line to the LCD.

If the LCD serial connection is left when programming command characters may be sent to the LCD and can change things such as backlight intensity so it is important to always disconnect the serial line when writing to the Arduino.

Connect the battery as shown in Appendix B and ensure that the second line is displaying a numerical value after "RSSI:"

The display shows the callsign you're tracking on the top line and the signal strength in recieved signal strength intensity (RSSI). RSSI is an exponential scale with a maximum value of 240.

5 Yagi

The yagi design in Appendix E is a 3-element antenna that uses two passive elements to shape the electric field and provide directional signal reception with the highest gain in line with the centerline of the antenna on the narrower side. The middle element is a dipole meaning that the antenna is split in half with one side connected to ground and the other connected to the antenna input on the radio.

In order to construct the dipolar element cut the element to length and then split it in half. Insulate the two halves from each other and then connect each to part of a shielded cable. In a shielded cable the wire mesh around the outside is the ground conductor and the wire inside is the signal conductor.

To use this antenna with the receiver connect the ground to the arduino ground terminal and the signal wire to the antenna pin of the rfm22b.

6 Usage

When you are close to a likely landing location sweep the horizon slowly with the yagi until you see a signal. Find the direction with the highest RSSI reading and go in that direction. As you get closer and the RSSI readings get closer to 240 the readings will begin to change faster with angle which should allow you to hone in on the location easier.