Ted Krimmer

Ranger Ripper plan is to create a way for the messages captured by the ranger from the Remus 100 to be sent to an Arduino computer then displayed and passed to the mission laptop simultaneously.

Ranger Ripper

Project Development

# Project Development – Ranger Ripper

Ted Krimmer – 12/01/2002

Contents

[Project Development – Ranger Ripper 1](#_Toc128386719)

[1. Introduction 2](#_Toc128386720)

[2. Software 2](#_Toc128386721)

[2.1. Python code 2](#_Toc128386722)

[2.2. Arduino code 2](#_Toc128386723)

[3. Hardware 3](#_Toc128386724)

# Introduction

The Ripper Ranger is a product designed to be attached to the REMUS Ranger unit. This has been designed for record keeping. The Ripper will automatically print out the relevant information received by acoustic communication from the Ranger along with the relevant GPS information of the Ripper’s position and the time the message is received. This system has been designed to be used with the REMUS 100 AUV (Automatic Underwater Vehicle)

This Product is broken into two parts. First will be the software. The second will be the hardware.

# Software

The software used in this project has 2 sections. The first section is the python code used for the OCR. The second is the Arduino sketch attached to the Ripper microcontroller. Please note that the python code isn’t finished as it became less relevant as the Ripper microcontroller was developed. The python code can successfully take a screenshot and transcribe the text to a log file. The implementation of the thermal printer was never finished. The python code is the file “Ripper\_Ranger”. This file also contains the code for the Arduino code. There are two Arduino folders named “Ripper\_Development\_Arduino” which has all the code from development, and “Ripper\_Ranger\_Arduino” which contains the final working version which is loaded onto the microcontroller.

## Python code

The Python code is found in the two folders. “Putty\_Ranger\_Python” contains 4 main scripts. “main.py” contains code that will track and save all messages sent/received through Putty. “main2.py” was used for testing the added future of saving the message if it is longer than 65 characters. “RangerMessage.py” Will take the message that is saved and formatted and saved as per the VIP software format. “RangerMessageTest.py” was for testing new functions.

The folder named “Ripper\_Ranger\_OCR\_Python” Contains the code for the OCR (On Screen Recorder) which is detailed in the development document. It contains a “main.py” which can be run for the program to begin.

## Arduino code

The Arduino code is in 2 folders. “Ripper\_Development\_Arduino” is all the Arduino sketches broken down into individual components. For example “Ranger\_printer\_beta(GPS)” contains just the code to run the GPS. This was used to test each part of the project for bugs. The “Ripper\_Ranger\_Arduino” folder has the final working version which is loaded onto the microcontroller.

# Hardware

The Arduino uno will be the microcontroller for getting the messages. The Arduino may use some additional attachments.

List of hardware used in the project.

* Arduino microcontroller– Uno R3
* Breadboard
* Thermal Printer – adafruit Mini Thermal Printer
* GPS module U-BLOX NEO-7M
* Arduino Compatible 128x64 Dot Matrix LCD Display Module
* RS-232 to TTL UART Converter Module
* RS-232 serial splitter cable

The information about the use of each piece of hardware can be found in the Development document.