**TITLE**

by  
Blaze D. Ethridge (BSME)  
Travis VanEastern (BSCE)

Theodore G. Cleveland

Final Report to …….

Texas Tech University

Month YYYY

# ABSTRACT

words

# ACKNOWLEDGEMENTS

# TABLE OF CONTENTS

**ABSTRACT i**

**ACKNOWLEDGEMENTS ii**

**TABLE OF CONTENTS iii**

**LIST OF FIGURES AND TABLES iv**

**CHAPTER 1 – INTRODUCTION 1**

**1.1 Motivation 1**

**1.2 Depth Measurement Value and Concepts 1**

**1.3 Report Structure 1**

**CHAPTER 2 – REVIEW OF PRIOR WORK 2**

**2.1 Depth Measurement for Safety Alert 2**

**2.2 Small Footprint Data Loggers 2**

**2.3 Interfaces 2**

**CHAPTER 3 – METHODS EMPLOYED 1**

**3.1 Conceptual System Design 1**

**3.2 Configure Raspberry Pi 3B+ as Wireless Host (Controller) 1**

**3.3 Configure Raspberry Pi Zero-W as Data Logger 1**

**3.4 Construct Analog-Digital Converter (A/D) 1**

**3.5 Modify Off-the-shelf Water Level Sensor to extend resolution 1**

**3.6 Use of 3-D Printing for Fabricating the Ladder Chassis 1**

**3.7 Python Script to Make Water Level Measurements 1**

**3.8 System Integration for Testing 1**

**CHAPTER 4 – RESULTS 1**

**4.1 Proof-of-Principle Prototype Testing 1**

**4.2 Deployment Prototype Testing 1**

**4.3 3D-Printed Chassis Testing 1**

**CHAPTER 5 – CONCLUSIONS 1**

**REFERENCES 1**

**APPENDIX A: GENERATE EBD VALUES FROM NWS PFDS 1**

**APPENDIX B: R-SCRIPT TO GENERATE IDF ESTIMATE FROM EBDUSA 5**

**APPENDIX C: WEB-BROWSER INTERFACE (HTML) 5**

**APPENDIX C: CGI-BIN PYTHON SCRIPT; GENERATE IDF FROM EBDUSA 1**

**APPENDIX D: PYTHON SCRIPT TO CALL GENERATE IDF FROM EBDUSA 1**

# LIST OF FIGURES AND TABLES



*Figure 1: Intensity-duration-frequency curve of maximum rainfall in Chicago, U.S.A.* ***Error! Bookmark not defined.***

*Figure 2: Depth-frequency Atlas for 2-year storm for 1-hour duration from 2004 DDF Atlas (Asquith).* ***Error! Bookmark not defined.***

*Figure 3: Example of acceptable .csv file for Code1 with duration in column A and ARI on row 14.* ***Error! Bookmark not defined.***

*Figure 4: Example of acceptable .csv file for Code2 with county in column A and duration on row 1.* ***Error! Bookmark not defined.***

*Figure 5: Process flowchart of R script for computing IDF coefficients* ***Error! Bookmark not defined.***

*Table 1: Error between OLS and NLM method for the 2 and 100-year ARI. 1*

# CHAPTER 1 – INTRODUCTION

words

## 1.1 Motivation

words

## 1.2 Depth Measurement Value and Concepts

words

## 1.3 Report Structure

# CHAPTER 2 – REVIEW OF PRIOR WORK

words

## 2.1 Depth Measurement for Safety Alert

words

## 2.2 Small Footprint Data Loggers

words

## 2.3 Interfaces

words

# CHAPTER 3 – METHODS EMPLOYED

## 3.1 Conceptual System Design

words

## 3.2 Configure Raspberry Pi 3B+ as Wireless Host (Controller)

Wireless host setup

Central controller

Digital level readout

## 3.3 Configure Raspberry Pi Zero-W as Data Logger

Python script to read the depth

## 3.4 Construct Analog-Digital Converter (A/D)

Words

## 3.5 Modify Off-the-shelf Water Level Sensor to extend resolution

Words

## 3.6 Use of 3-D Printing for Fabricating the Ladder Chassis

## 3.7 Python Script to Make Water Level Measurements

Words + put a copy of the script here

## 3.8 System Integration for Testing

# 

# CHAPTER 4 – RESULTS

words

## 4.1 Proof-of-Principle Prototype Testing

Picture of the original prototype.

Graph of the Readings vs Depth

Manual testing by shorting

## 4.2 Deployment Prototype Testing

Picture of the prototype

Graph of readings versus depth

Assessment of linearity

## 4.3 3D-Printed Chassis Testing

Picture of 3D printed

Repeatability (from a fabrication standpoint)

# CHAPTER 5 – CONCLUSIONS

words

# REFERENCES

1. SunFounder Inc. 2016. “Super Kit V2.0 for Raspberry Pi – Instruction Manual” ShenZhen Maker Stars Tech. Co., Ltd. D507 HuaChuangDa Science&Culture Industry Park, Bao’An District, HaiHui Road, ShanZhen, GuanDong, 518100 CHINA. 72p.

# APPENDIX A: GENERATE EBD VALUES FROM NWS PFDS

# RCode finds rainfall coefficients for each ARI based on IDF model and nlm package

# Single county

# Adapted from Tay, C. C. 2015. DEVELOPING IDF MODELS USING NONLINEAR MINIMIZATION IN R

#

#### SETUP ####

(WD <- getwd()) #gets working directory

if (!is.null(WD)) setwd(WD) #sets working directory

rm(list = ls()) #removes variables from workspace

#detach(ddf) #preventative measure to detach old excel files

# VERIFY RUNS OK TO HERE 5-SEP-17 TGC

# Assumes THIS script is in same directory as the input source and target output

#########################

## PROTOTYPE FUNCTIONS ##

#########################

depthfunc <- function(tc,eee,bee,dee) #Depth model - tc is time of concentration in minutes

{

dep <- (tc/60)\*(bee/((dee+tc)^eee))

return(dep)

}

#########################

intensityfunc <- function(tc,eee,bee,dee) #Intensity model - tc is time of concentration in minutes

{

int <- bee/((dee+tc)^eee) #in/hr

return(int)

}

#########################

sse <- function(x,duration,ddfdepth) #Sum of Errors Squared Function (SSE) used in NLM process

{

# x[1] == eee; x[2] == bee; x[3] == dee

sum((ddfdepth - depthfunc(duration,x[1],x[2],x[3]))^2)

}

#########################

############ READ THE FILE FROM PFDS ##############################################

### future version make this interactive so user can enter the filename from console

#ddf <- read.csv("PF\_Depth\_English\_AMS.csv", header = FALSE,na.strings = "")

ddf <- file("PF\_Depth\_English\_AMS.csv", "r") #make connection to input file

metadata <- (readLines(ddf, n=14, ok = TRUE, warn = TRUE,encoding = "unknown", skipNul = FALSE))

durfreq <- (readLines(ddf, n=10, ok = TRUE, warn = TRUE,encoding = "unknown", skipNul = FALSE))

close(ddf) # close the connection

# metadata is first 14 rows of the file -- these are ignored

# split the string

durfreq <- unlist(strsplit(durfreq,split=","))

# durfreq is the next 10 rows of data

durat <- character(0)

depth2 <- numeric(0)

depth5 <- numeric(0)

depth10 <- numeric(0)

depth25 <- numeric(0)

depth50 <- numeric(0)

depth100 <- numeric(0)

depth200 <- numeric(0)

depth500 <- numeric(0)

depth1000 <- numeric(0)

irow <- -9

for(i in 1:10){

irow <- irow+10

durat[i] <- durfreq[irow] # this is a string array

depth2[i] <- as.numeric(durfreq[irow+1]) # convert to numeric

depth5[i] <- as.numeric(durfreq[irow+2]) # convert to numeric

depth10[i] <- as.numeric(durfreq[irow+3]) # convert to numeric

depth25[i] <- as.numeric(durfreq[irow+4]) # convert to numeric

depth50[i] <- as.numeric(durfreq[irow+5]) # convert to numeric

depth100[i] <- as.numeric(durfreq[irow+6]) # convert to numeric

depth200[i] <- as.numeric(durfreq[irow+7]) # convert to numeric

depth500[i] <- as.numeric(durfreq[irow+8]) # convert to numeric

depth1000[i] <- as.numeric(durfreq[irow+9]) # convert to numeric

}

# Force 10min = 5min values

# first save the 5 min for later plotting

d2temp <- depth2[1]

d5temp <- depth5[1]

d10temp <- depth10[1]

d25temp <- depth25[1]

d50temp <- depth50[1]

d100temp <- depth100[1]

# Minimization Process

duration <- c(5/60,10/60,15/60,30/60,1,2,3,6,12,24) # durations, numeric in hours

duration <- 60\*duration # convert into minutes to work with the prototype functions

# nlm starting guess vector

x <- vector()

x[1] <- 1.004 #eee

x[2] <- 1.004 #bee

x[3] <- 1.004 #dee

NEW <- nlm(sse,c(x[1],x[2],x[3]),duration,depth2,steptol=1e-16, gradtol=1e-6)

ebd2 <- NEW$estimate

depthmodel2 <- depthfunc(duration,ebd2[1],ebd2[2],ebd2[3])

x <- NEW$estimate #use these values for next ARI

#

NEW <- nlm(sse,c(x[1],x[2],x[3]),duration,depth5,steptol=1e-16, gradtol=1e-6)

ebd5 <- NEW$estimate

depthmodel5 <- depthfunc(duration,ebd5[1],ebd5[2],ebd5[3])

x <- NEW$estimate #use these values for next ARI

#

NEW <- nlm(sse,c(x[1],x[2],x[3]),duration,depth10,steptol=1e-16, gradtol=1e-6)

ebd10 <- NEW$estimate

depthmodel10 <- depthfunc(duration,ebd10[1],ebd10[2],ebd10[3])

x <- NEW$estimate #use these values for next ARI

#

NEW <- nlm(sse,c(x[1],x[2],x[3]),duration,depth25,steptol=1e-16, gradtol=1e-6)

ebd25 <- NEW$estimate

depthmodel25 <- depthfunc(duration,ebd25[1],ebd25[2],ebd25[3])

x <- NEW$estimate #use these values for next ARI

#

NEW <- nlm(sse,c(x[1],x[2],x[3]),duration,depth50,steptol=1e-16, gradtol=1e-6)

ebd50 <- NEW$estimate

depthmodel50 <- depthfunc(duration,ebd50[1],ebd50[2],ebd50[3])

x <- NEW$estimate #use these values for next ARI

#

NEW <- nlm(sse,c(x[1],x[2],x[3]),duration,depth100,steptol=1e-16, gradtol=1e-6)

ebd100 <- NEW$estimate

depthmodel100 <- depthfunc(duration,ebd100[1],ebd100[2],ebd100[3])

outfile <- file("output.txt","w")

write(c(ebd2[1],ebd2[2],ebd2[3]),outfile,sep=",")

write(c(ebd5[1],ebd5[2],ebd5[3]),outfile,sep=",")

write(c(ebd10[1],ebd10[2],ebd10[3]),outfile,sep=",")

write(c(ebd25[1],ebd25[2],ebd25[3]),outfile,sep=",")

write(c(ebd50[1],ebd50[2],ebd50[3]),outfile,sep=",")

write(c(ebd100[1],ebd100[2],ebd100[3]),outfile,sep=",")

close(outfile)

# plotting for QA/QC checks

# now put the 5 minute values back onto the vectors for plotting

depth2[1] <- d2temp

depth5[1] <- d5temp

depth10[1] <- d10temp

depth25[1] <- d25temp

depth50[1] <- d50temp

depth100[1] <- d100temp

plot(duration,depth100,log="xy",ylim=c(0.1,12),ylab="Depth (inches)",xlab="Duration (minutes)")

lines(duration,depthmodel100,col="blue")

lines(duration,depth50,type="p")

lines(duration,depthmodel50,col="blue")

lines(duration,depth25,type="p")

lines(duration,depthmodel25,col="blue")

lines(duration,depth10,type="p")

lines(duration,depthmodel10,col="blue")

lines(duration,depth5,type="p")

lines(duration,depthmodel5,col="blue")

lines(duration,depth2,type="p")

lines(duration,depthmodel2,col="blue")

# APPENDIX B: R-SCRIPT TO GENERATE IDF ESTIMATE FROM EBDUSA

# APPENDIX C: WEB-BROWSER INTERFACE (HTML)

# APPENDIX C: CGI-BIN PYTHON SCRIPT; GENERATE IDF FROM EBDUSA

# APPENDIX D: PYTHON SCRIPT TO CALL GENERATE IDF FROM EBDUSA