**RASPIGUARD**

From: Vivek Socrates & Karel Tutsu  
Discipline: Computer Engineering Technology  
Date: N.A. / N.A. / 2018

# Declaration of Joint Authorship

Vivek Socrates, Karel Tutsu and Heakeme Williams confirm that this project is a joint collection of work between the three of us. Any and all outside sources that have been referenced or used will be cited and properly acknowledged at point of use.

# Approved Proposal

## Executive Summary

<TO BE FILLED>

## Background

<TO BE FILLED>

## Concluding remarks

<TO BE FILLED>

# Abstract

The problem solved by this project is installing a home security system can be costly, but needing one and not having one can cost you more. Fortunately, you will not have that dilemma with RaspiGuard. RaspiGuard is a cost-effective and easy to install security system that you can easily setup and deploy on your own. A bit of background about this topic is RaspiGuard is a home security/surveillance system that is used to monitor a room remotely using our android application. The current operational functionality is door and moisture sensors. We are currently working on the functionality of light sensors as well as various other sensors.

Table of Contents

[Declaration of Joint Authorship iii](#_Toc506214988)

[Approved Proposal v](#_Toc506214989)

[Executive Summary v](#_Toc506214990)

[Background v](#_Toc506214991)

[Concluding remarks v](#_Toc506214992)

[Abstract vii](#_Toc506214993)

[1. Introduction 13](#_Toc506214994)

[2. Project Description 15](#_Toc506214995)

[2.1 Problem 15](#_Toc506214996)

[2.2 Rationale Behind Project 15](#_Toc506214997)

[2.3 Project Scope 15](#_Toc506214998)

[2.4 Requirements Specifications 15](#_Toc506214999)

[2.4.1 Database 15](#_Toc506215000)

[2.4.2 Mobile Application 15](#_Toc506215001)

[2.4.3 Web Interface 15](#_Toc506215002)

[2.4.4 Networked Platform Communication Software 15](#_Toc506215003)

[2.4.5 Microprocessor Firmware 16](#_Toc506215004)

[2.4.6 Hardware Components 16](#_Toc506215005)

[2.5 Project Overview 16](#_Toc506215006)

[2.5.1 Bill of Materials 16](#_Toc506215007)

[2.5.2 Time Commitment 16](#_Toc506215008)

[2.5.3 Mechanical Assembly 16](#_Toc506215009)

[2.5.4 PCB and Soldering 16](#_Toc506215010)

[2.5.5 Power Up 17](#_Toc506215011)

[2.5.6 Unit Testing 17](#_Toc506215012)

[2.5.7 Production Testing 17](#_Toc506215013)

[2.6 Problems Encountered 17](#_Toc506215014)

[2.6.1 <TO BE FILLED> 17](#_Toc506215015)

[2.6.2 <TO BE FILLED> 17](#_Toc506215016)

[2.6.3 <TO BE FILLED> 17](#_Toc506215017)

[2.6.4 <TO BE FILLED> 17](#_Toc506215018)

[2.6.5 <TO BE FILLED> 17](#_Toc506215019)

[2.7 Approaches 17](#_Toc506215020)

[2.7.1 <TO BE FILLED> 17](#_Toc506215021)

[2.7.2 <TO BE FILLED> 17](#_Toc506215022)

[2.7.3 <TO BE FILLED> 17](#_Toc506215023)

[2.7.4 <TO BE FILLED> 17](#_Toc506215024)

[2.7.5 <TO BE FILLED> 17](#_Toc506215025)

[2.8 Walkthrough of System 18](#_Toc506215026)

[2.8.1 Microcontroller 18](#_Toc506215027)

[2.8.2 Microprocessor 18](#_Toc506215028)

[2.8.3 Server 18](#_Toc506215029)

[2.8.4 Android Phone Application 18](#_Toc506215030)

[2.8.5 Website 18](#_Toc506215031)

[3. Progress Reports 19](#_Toc506215032)

[3.1 Report 1 19](#_Toc506215033)

[3.2 Report 2 19](#_Toc506215034)

[3.4 Report 4 19](#_Toc506215035)

[3.5 Report 5 19](#_Toc506215036)

[3.6 Report 6 19](#_Toc506215037)

[4. Conclusions 21](#_Toc506215038)

[5. Recommendations 23](#_Toc506215039)

[6. Technical References 25](#_Toc506215040)

[7. Appendices 27](#_Toc506215041)

[7.1 Microcontroller Firmware 27](#_Toc506215042)

[7.2 Microprocessor Communication Script 27](#_Toc506215043)

[7.3 Microprocessor Database Communication Program 27](#_Toc506215044)

[7.4 Database Input Script 27](#_Toc506215045)

[7.5 Database Retrieval Script for Phone Application 27](#_Toc506215046)

[7.6 Website Code 27](#_Toc506215047)

[7.6.1 <TO BE FILLED> 27](#_Toc506215048)

[7.6.2 <TO BE FILLED> 27](#_Toc506215049)

[7.6.3 <TO BE FILLED> 27](#_Toc506215050)

[7.6.4 <TO BE FILLED> 27](#_Toc506215051)

[7.6.5 <TO BE FILLED> 27](#_Toc506215052)

[7.6.6 <TO BE FILLED> 27](#_Toc506215053)

[7.7 Android Phone Application 27](#_Toc506215054)

[7.7.1 <TO BE FILLED> 27](#_Toc506215055)

[7.7.2 <TO BE FILLED> 27](#_Toc506215056)

[7.7.3 <TO BE FILLED> 28](#_Toc506215057)

[7.7.4 <TO BE FILLED> 28](#_Toc506215058)

[7.7.5 <TO BE FILLED> 28](#_Toc506215059)

[7.7.6 <TO BE FILLED> 28](#_Toc506215060)

[7.7.7 <TO BE FILLED> 28](#_Toc506215061)

[7.7.8 <TO BE FILLED> 28](#_Toc506215062)

[7.7.9 <TO BE FILLED> 28](#_Toc506215063)

[7.7.10 <TO BE FILLED> 28](#_Toc506215064)

[7.7.11 <TO BE FILLED> 28](#_Toc506215065)

[7.7.12 <TO BE FILLED> 28](#_Toc506215066)

[7.7.13 <TO BE FILLED> 28](#_Toc506215067)

[7.7.14 <TO BE FILLED> 28](#_Toc506215068)

[7.7.15 <TO BE FILLED> 28](#_Toc506215069)

[7.7.16 <TO BE FILLED> 28](#_Toc506215070)

[7.7.17 <TO BE FILLED> 29](#_Toc506215071)

[7.7.18 <TO BE FILLED> 29](#_Toc506215072)

[7.7.19 <TO BE FILLED> 29](#_Toc506215073)

[7.7.20 <TO BE FILLED> 29](#_Toc506215074)

[7.7.21 <TO BE FILLED> 29](#_Toc506215075)

[7.7.22 <TO BE FILLED> 29](#_Toc506215076)

[7.7.23 <TO BE FILLED> 29](#_Toc506215077)

[7.7.24 <TO BE FILLED> 29](#_Toc506215078)

# 1. Introduction

RaspiGuard is a home security/surveillance system that is used to monitor a room remotely using our android application. The current operational functionality is door and moisture sensors. We are currently working on the functionality of light sensors as well as various other sensors. Installing a home security system can be costly, but needing one and not having one can cost you more. Fortunately, you will not have that dilemma with RaspiGuard. RaspiGuard is a cost-effective and easy to install security system that you can easily setup and deploy on your own.

# 2. Project Description

## 2.1 Problem

<TO BE FILLED>

## 2.2 Rationale Behind Project

<TO BE FILLED>

## 2.3 Project Scope

<TO BE FILLED>

## 2.4 Requirements Specifications

### 2.4.1 Database

In our application we will be using the standard MySQL (version 5.0.27). The database will store user account information, the live status of sensors and logs sensor activity.

All the commands and table operations are very basic such as INSERT, SELECT, UPDATE queries which are supported in earlier versions as well.

There will be a separate table for users, activity log and every sensor will have their own separate table in the database. Karel will lead this development.

### 2.4.2 Mobile Application

Our Android mobile application will be the primary end user interface to manipulate with sensors and data. The core functionality will include:

* Ability to add, modify or remove sensors from their account
* Ability to view the current status of a chosen sensor
* Ability to toggle sensor alarm feature ON/OFF
* Ability view the event history (activity log) of the sensor
* Ability to change user account information

Karel will lead this development.

### 2.4.3 Web Interface

The web interface will serve as a secondary interface to the user – the main benefit being portability. The user will be able to do all the same things as in the mobile application above just simply in a web browser. Karel will lead this development.

### 2.4.4 Networked Platform Communication Software

On the Broadcom BCM2837 (Raspberry Pi) device there will be software that will connect to an MySQL database hosted on cloud services provider [*DigitalOcean*](https://www.digitalocean.com/)*.* The device will be authenticated using credentials stored on the database and will update sensor fields at constant regular intervals. Vivek will lead this development.

### 2.4.5 Microprocessor Firmware

On the Broadcom device, there is a python script that will read raw data from analog and digital sensors. The data will be converted, formatted, and displayed using the standard measurement for each individual type of sensor. Moisture sensors will display humidity in percentage, door sensors will show status as “Open” or “Closed”, and light sensors will display luminous emittance in lux values. Vivek will lead this development.

### 2.4.6 Hardware Components

Components include:

* Raspberry Pi 3 Model B
* Raspberry Pi Power Adapter
* adafruit PiTFT 3.5” Touchscreen
* Magnetic Contact Switch
* SparkFun Moisture Sensor
* Photoresistor
* adafruit ADS1115
* MicroSD card
* Laser Cut Housing / Case

These components will be cased inside the laser cut housing. The components will be inaccessible to the consumer and will power up and start-up the custom written software upon the board being powered. HDMI, USB, audio, and Ethernet ports will be blocked, disabled and inaccessible. The sensors will be attached to the PCB that will be inside the case. Vivek will lead this development.

## 2.5 Project Overview

### 2.5.1 Bill of Materials

<TO BE FILLED>

### 2.5.2 Time Commitment

<TO BE FILLED>

### 2.5.3 Mechanical Assembly

<TO BE FILLED>

### 2.5.4 PCB and Soldering

<TO BE FILLED>

### 2.5.5 Power Up

<TO BE FILLED>

### 2.5.6 Unit Testing

<TO BE FILLED>

### 2.5.7 Production Testing

<TO BE FILLED>

## 2.6 Problems Encountered

### 2.6.1 <TO BE FILLED>

<TO BE FILLED>

### 2.6.2 <TO BE FILLED>

<TO BE FILLED>

### 2.6.3 <TO BE FILLED>

<TO BE FILLED>

### 2.6.4 <TO BE FILLED>

<TO BE FILLED>

### 2.6.5 <TO BE FILLED>

<TO BE FILLED>

## 2.7 Approaches

### 2.7.1 <TO BE FILLED>

<TO BE FILLED>

### 2.7.2 <TO BE FILLED>

<TO BE FILLED>

### 2.7.3 <TO BE FILLED>

<TO BE FILLED>

### 2.7.4 <TO BE FILLED>

<TO BE FILLED>

### 2.7.5 <TO BE FILLED>

<TO BE FILLED>

## 2.8 Walkthrough of System

### 2.8.1 Microcontroller

<TO BE FILLED>

### 2.8.2 Microprocessor

<TO BE FILLED>

### 2.8.3 Server

<TO BE FILLED>

### 2.8.4 Android Phone Application

<TO BE FILLED>

### 2.8.5 Website

<TO BE FILLED>

# 3. Progress Reports

## 3.1 Report 1

<TO BE FILLED>

## 3.2 Report 2

<TO BE FILLED>

## 3.4 Report 4

<TO BE FILLED>

## 3.5 Report 5

<TO BE FILLED>

## 3.6 Report 6

<TO BE FILLED>

# 4. Conclusions

<TO BE FILLED>

# 5. Recommendations

<TO BE FILLED>

# 6. Technical References

<TO BE FILLED>

# 7. Appendices

## 7.1 Microcontroller Firmware

<TO BE FILLED>

## 7.2 Microprocessor Communication Script

<TO BE FILLED>

## 7.3 Microprocessor Database Communication Program

<TO BE FILLED>

## 7.4 Database Input Script

<TO BE FILLED>

## 7.5 Database Retrieval Script for Phone Application

<TO BE FILLED>

## 7.6 Website Code

### 7.6.1 <TO BE FILLED>

<TO BE FILLED>

### 7.6.2 <TO BE FILLED>

<TO BE FILLED>

### 7.6.3 <TO BE FILLED>

<TO BE FILLED>

### 7.6.4 <TO BE FILLED>

<TO BE FILLED>

### 7.6.5 <TO BE FILLED>

<TO BE FILLED>

### 7.6.6 <TO BE FILLED>

<TO BE FILLED>

## 7.7 Android Phone Application

### 7.7.1 <TO BE FILLED>

<TO BE FILLED>

### 7.7.2 <TO BE FILLED>

<TO BE FILLED>

### 7.7.3 <TO BE FILLED>

<TO BE FILLED>

### 7.7.4 <TO BE FILLED>

<TO BE FILLED>

### 7.7.5 <TO BE FILLED>

<TO BE FILLED>

### 7.7.6 <TO BE FILLED>

<TO BE FILLED>

### 7.7.7 <TO BE FILLED>

<TO BE FILLED>

### 7.7.8 <TO BE FILLED>

<TO BE FILLED>

### 7.7.9 <TO BE FILLED>

<TO BE FILLED>

### 7.7.10 <TO BE FILLED>

<TO BE FILLED>

### 7.7.11 <TO BE FILLED>

<TO BE FILLED>

### 7.7.12 <TO BE FILLED>

<TO BE FILLED>

### 7.7.13 <TO BE FILLED>

<TO BE FILLED>

### 7.7.14 <TO BE FILLED>

<TO BE FILLED>

### 7.7.15 <TO BE FILLED>

<TO BE FILLED>

### 7.7.16 <TO BE FILLED>

<TO BE FILLED>

### 7.7.17 <TO BE FILLED>

<TO BE FILLED>

### 7.7.18 <TO BE FILLED>

<TO BE FILLED>

### 7.7.19 <TO BE FILLED>

<TO BE FILLED>

### 7.7.20 <TO BE FILLED>

<TO BE FILLED>

### 7.7.21 <TO BE FILLED>

<TO BE FILLED>

### 7.7.22 <TO BE FILLED>

<TO BE FILLED>

### 7.7.23 <TO BE FILLED>

<TO BE FILLED>

### 7.7.24 <TO BE FILLED>

<TO BE FILLED>