RASPIGUARD

Submitted By: Vivek Socrates, Karel Tutsu, & Heakeme Williams  
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.

# Proposal For Development Of RaspiGuard

Prepared by Vivek Socrates & Karel Tutsu

Computer Engineering Technology Students

<https://github.com/V-Socrates/RaspiGuard>

# Executive Summary

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet-connected hardware will include a custom PCB with the following sensors and actuators, moisture sensor and magnetic contact switches. The database will store the status of the sensors, log activity information, and will be used to store user accounts for application authentication. The mobile device functionality will include the ability to monitor sensor state, change alarm settings, create and modify account information, add and remove monitoring units, view activity logs, as well as receiving notifications of sensor activity, and will be further detailed in the mobile application proposal. I will be collaborating with Vlad and Kelly in the prototype lab regarding the housing and deployment of the system. In the winter semester, I plan to form a group with the following students, who are also building similar hardware this term and working on the mobile application with me Vivek Socrates, Karel Tutsu and Heakeme Williams. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a group of three.

# Background

The problem solved by RaspiGuard is that installing a home security system can be costly, but needing one and not having one can cost you even 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. 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.

Existing products on the market include [1]. I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content”[2] and have found and read [3] which provides insight into similar efforts.

In the Computer Engineering Technology program, we have learned about the following topics from the respective relevant courses:

* Java Docs from CENG 212 Programming Techniques In Java,
* Construction of circuits from CENG 215 Digital And Interfacing Systems,
* Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
* Micro computing from CENG 252 Embedded Systems,
* SQL from CENG 254 Database With Java,
* Web access of databases from CENG 256 Internet Scripting; and,
* Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

# Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:

## Phase 1 Hardware build

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

## Phase 2 System integration

The system integration will be completed in the fall term.

## Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that I have learned to potential employers. The brief description below provides rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines. New moisture sensor due to current moisture sensor reading being imprecise.

# Concluding remarks

This proposal presents a plan for providing an IoT solution for Our product will be a small unit that is wall mountable and can be used to monitor door activity, moisture of plants and other various information that is currently being worked on. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by [3]. I request approval of this project.

# 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.

# 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.

Contents

[Declaration of Joint Authorship II](#_Toc508991989)

[Proposal For Development Of RaspiGuard iii](#_Toc508991990)

[Executive Summary iii](#_Toc508991991)

[Background iv](#_Toc508991992)

[Methodology v](#_Toc508991993)

[Phase 1 Hardware build v](#_Toc508991994)

[Phase 2 System integration v](#_Toc508991995)

[Phase 3 Demonstration to future employers v](#_Toc508991996)

[Concluding remarks vi](#_Toc508991997)

[Abstract vii](#_Toc508991998)

[1. Introduction viii](#_Toc508991999)

[2. Project Description 15](#_Toc508992000)

[2.1 Problem 15](#_Toc508992001)

[2.2 Rationale Behind Project 15](#_Toc508992002)

[2.3 Project Scope 15](#_Toc508992003)

[2.4 Requirements Specifications 15](#_Toc508992004)

[2.4.1 Database 15](#_Toc508992005)

[2.4.2 Mobile Application 15](#_Toc508992006)

[2.4.3 Web Interface 16](#_Toc508992007)

[2.4.4 Networked Platform Communication Software 16](#_Toc508992008)

[2.4.5 Microprocessor Firmware 16](#_Toc508992009)

[2.4.6 Hardware Components 16](#_Toc508992010)

[2.5 Project Overview 17](#_Toc508992011)

[2.5.1 Bill of Materials 17](#_Toc508992012)

[2.5.2 Time Commitment 17](#_Toc508992013)

[2.5.3 Mechanical Assembly 17](#_Toc508992014)

[2.5.4 PCB and Soldering 17](#_Toc508992015)

[2.5.5 Power Up 17](#_Toc508992016)

[2.5.6 Unit Testing 17](#_Toc508992017)

[2.5.7 Production Testing 17](#_Toc508992018)

[2.6 Problems Encountered 17](#_Toc508992019)

[2.6.1 <TO BE FILLED> 17](#_Toc508992020)

[2.6.2 <TO BE FILLED> 17](#_Toc508992021)

[2.6.3 <TO BE FILLED> 18](#_Toc508992022)

[2.6.4 <TO BE FILLED> 18](#_Toc508992023)

[2.6.5 <TO BE FILLED> 18](#_Toc508992024)

[2.7 Approaches 18](#_Toc508992025)

[2.7.1 <TO BE FILLED> 18](#_Toc508992026)

[2.7.2 <TO BE FILLED> 18](#_Toc508992027)

[2.7.3 <TO BE FILLED> 18](#_Toc508992028)

[2.7.4 <TO BE FILLED> 18](#_Toc508992029)

[2.7.5 <TO BE FILLED> 18](#_Toc508992030)

[2.8 Walkthrough of System 18](#_Toc508992031)

[2.8.1 Microcontroller 18](#_Toc508992032)

[2.8.2 Microprocessor 18](#_Toc508992033)

[2.8.3 Server 18](#_Toc508992034)

[2.8.4 Android Phone Application 18](#_Toc508992035)

[2.8.5 Website 19](#_Toc508992036)

[3. Progress Reports 21](#_Toc508992037)

[3.1 Report 1 21](#_Toc508992038)

[3.2 Report 2 23](#_Toc508992039)

[3.4 Report 4 23](#_Toc508992040)

[3.5 Report 5 23](#_Toc508992041)

[3.6 Report 6 23](#_Toc508992042)

[4. Conclusions 25](#_Toc508992043)

[5. Recommendations 27](#_Toc508992044)

[6. Technical References 29](#_Toc508992045)

[7. Appendices 31](#_Toc508992046)

[7.1 Microcontroller Firmware 31](#_Toc508992047)

[7.2 Microprocessor Communication Script 31](#_Toc508992048)

[7.3 Microprocessor Database Communication Program 31](#_Toc508992049)

[7.4 Database Input Script 31](#_Toc508992050)

[7.5 Database Retrieval Script for Phone Application 31](#_Toc508992051)

[7.6 Website Code 31](#_Toc508992052)

[7.6.1 <TO BE FILLED> 31](#_Toc508992053)

[7.6.2 <TO BE FILLED> 31](#_Toc508992054)

[7.6.3 <TO BE FILLED> 31](#_Toc508992055)

[7.6.4 <TO BE FILLED> 31](#_Toc508992056)

[7.6.5 <TO BE FILLED> 31](#_Toc508992057)

[7.6.6 <TO BE FILLED> 31](#_Toc508992058)

[7.7 Android Phone Application 31](#_Toc508992059)

[7.7.1 <TO BE FILLED> 31](#_Toc508992060)

[7.7.2 <TO BE FILLED> 32](#_Toc508992061)

[7.7.3 <TO BE FILLED> 32](#_Toc508992062)

[7.7.4 <TO BE FILLED> 32](#_Toc508992063)

[7.7.5 <TO BE FILLED> 32](#_Toc508992064)

[7.7.6 <TO BE FILLED> 32](#_Toc508992065)

[7.7.7 <TO BE FILLED> 32](#_Toc508992066)

[7.7.8 <TO BE FILLED> 32](#_Toc508992067)

[7.7.9 <TO BE FILLED> 32](#_Toc508992068)

[7.7.10 <TO BE FILLED> 32](#_Toc508992069)

[7.7.11 <TO BE FILLED> 32](#_Toc508992070)

[7.7.12 <TO BE FILLED> 32](#_Toc508992071)

[7.7.13 <TO BE FILLED> 32](#_Toc508992072)

[7.7.14 <TO BE FILLED> 32](#_Toc508992073)

[7.7.15 <TO BE FILLED> 33](#_Toc508992074)

[7.7.16 <TO BE FILLED> 33](#_Toc508992075)

[7.7.17 <TO BE FILLED> 33](#_Toc508992076)

[7.7.18 <TO BE FILLED> 33](#_Toc508992077)

[7.7.19 <TO BE FILLED> 33](#_Toc508992078)

[7.7.20 <TO BE FILLED> 33](#_Toc508992079)

[7.7.21 <TO BE FILLED> 33](#_Toc508992080)

[7.7.22 <TO BE FILLED> 33](#_Toc508992081)

[7.7.23 <TO BE FILLED> 33](#_Toc508992082)

[7.7.24 <TO BE FILLED> 33](#_Toc508992083)

# 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

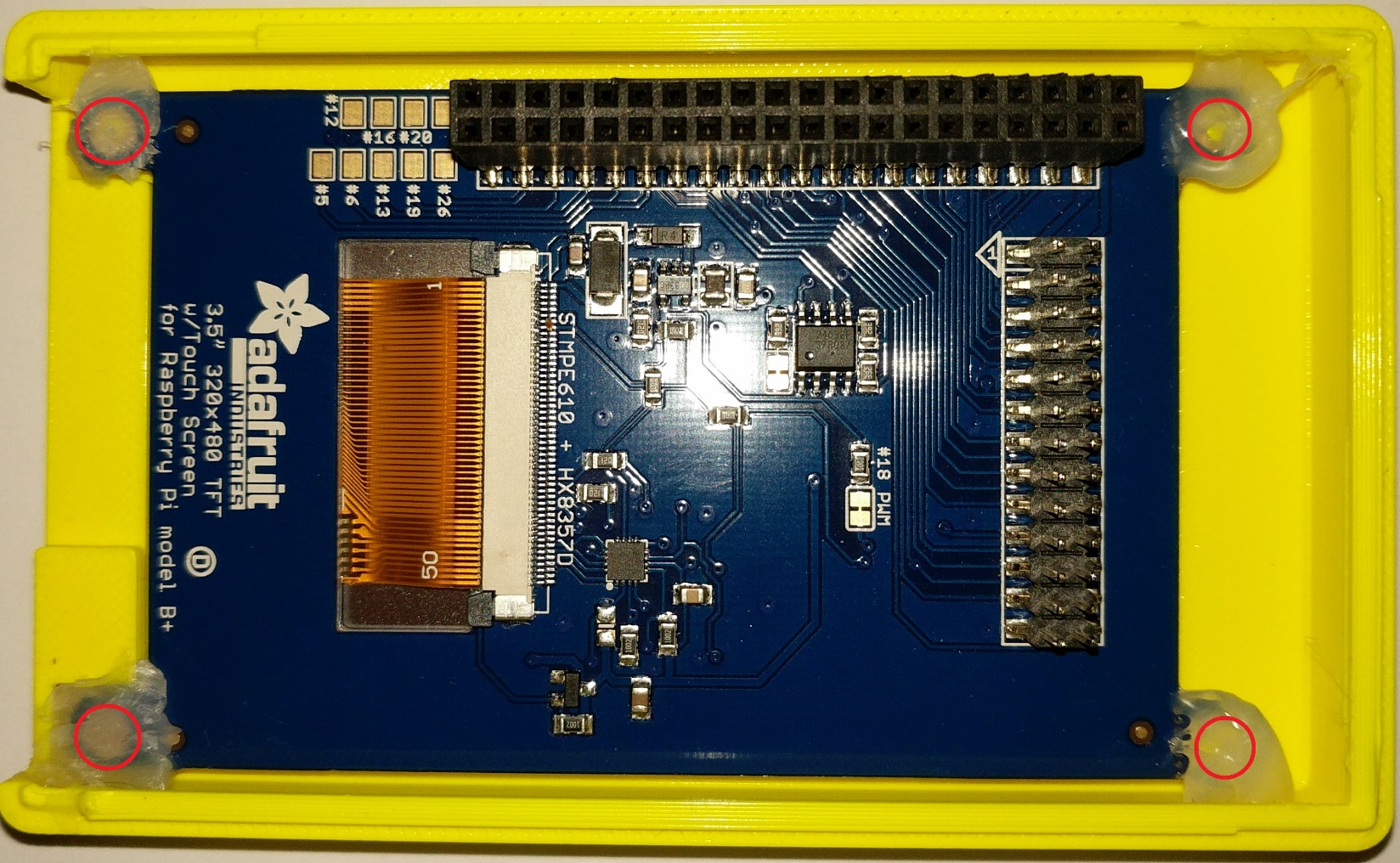
<TO BE FILLED>

### 2.5.3 Mechanical Assembly





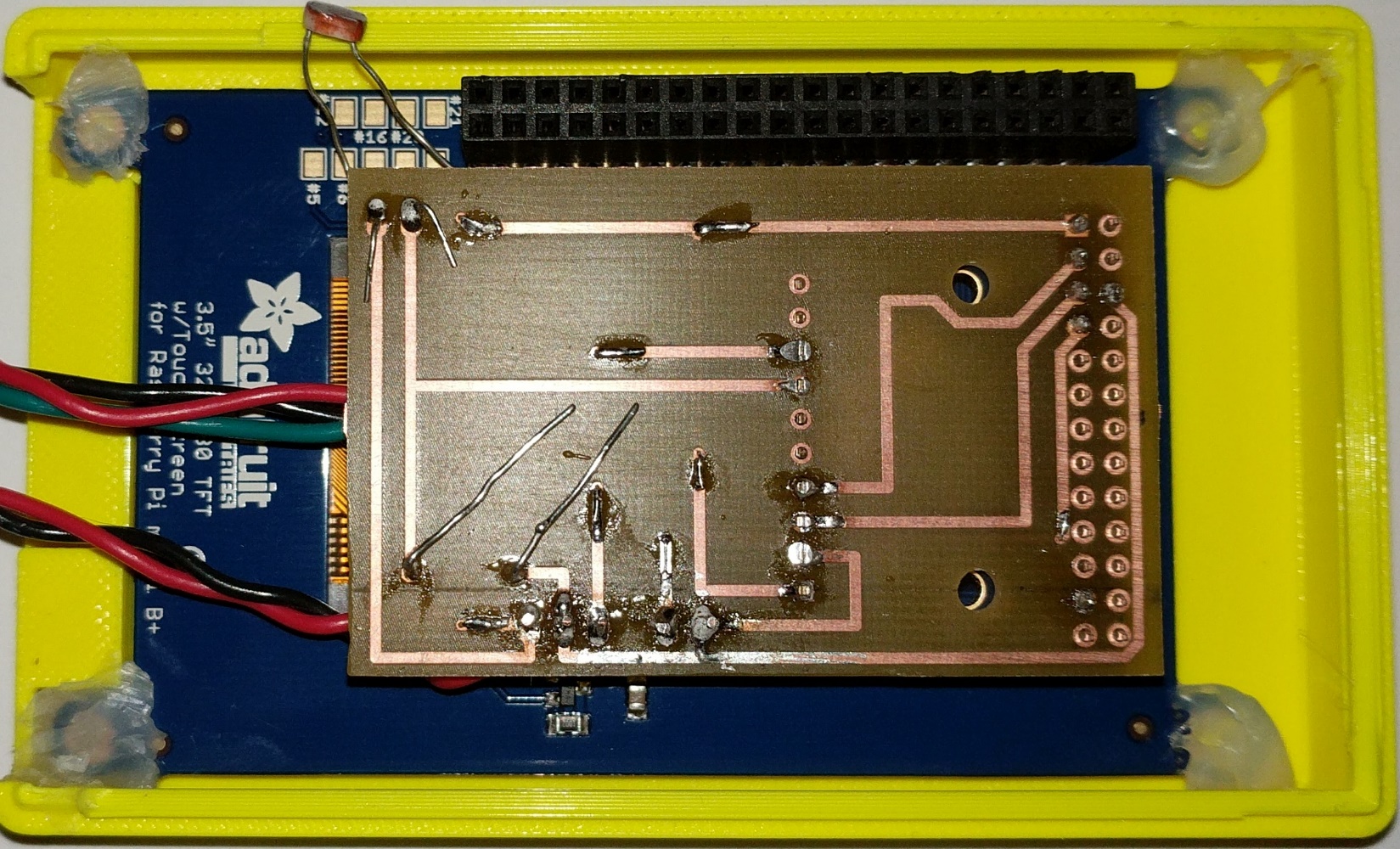
Drill small hole on the bottom side of the case in the space between the GPIO pins and the USB ports. Refer to above pictures.

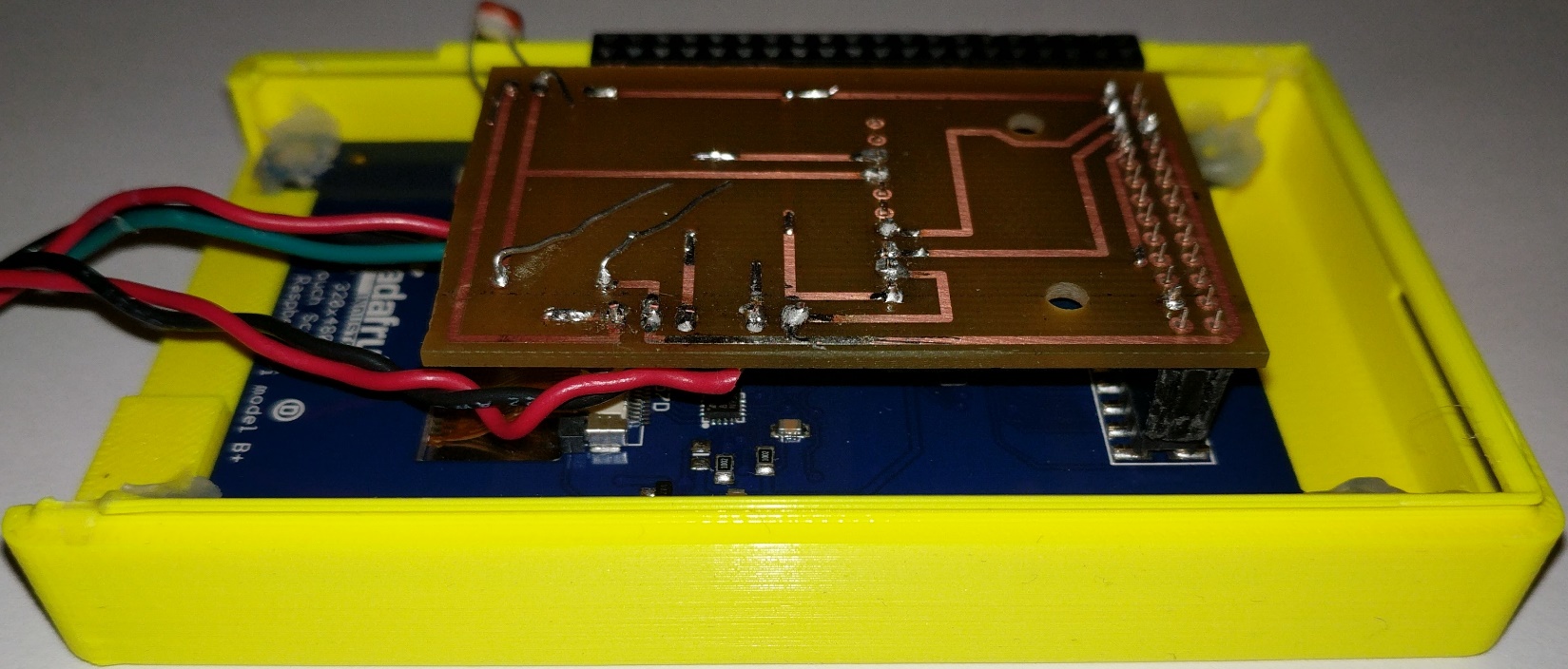


Hot glue the PiTFT to the top of the case.



Hot glue the Raspberry Pi to the bottom case.





Connect the PCB to the bottom of the PiTFT.



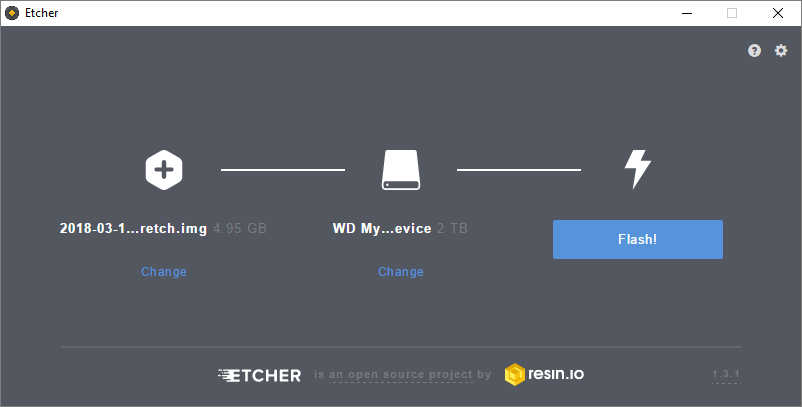
Connect the PiTFT to the Raspberry Pi and close the case. Make sure the photo resistor is aligned with the hole that was drilled.

### 2.5.4 PCB and Soldering

<TO BE FILLED>

### 2.5.5 Micro SD Card Setup

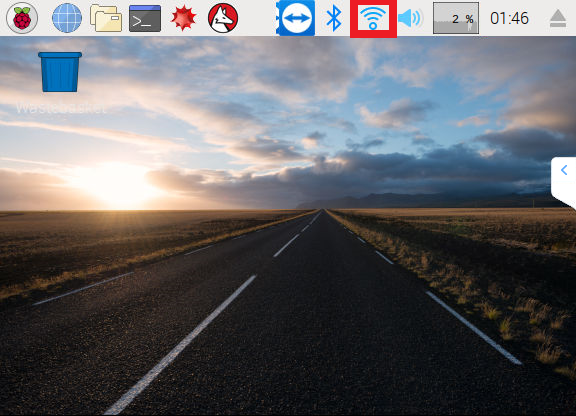
Download a program called [Etcher](https://etcher.io/) and the latest [Raspbian](https://www.raspberrypi.org/downloads/raspbian/) OS with Desktop.



Open etcher and click “Select image”. Navigate to where you Raspbian OS is located and select “Open”. Next click on “Select drive” and choose your Micro SD card and click “Continue”. After completing there steps click “Flash!” and wait for you Micro SD card to be flashed with the image. After completion insert your Micro SD card into the Raspberry Pi.

### 2.5.6 Rasbian Setup

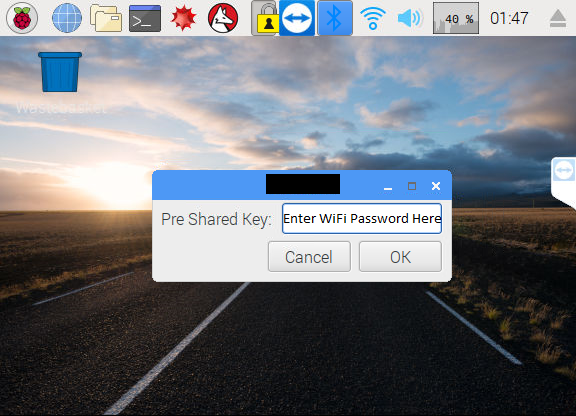
To setup the RaspiGuard unit plug a keyboard and mouse, then power on the unit.



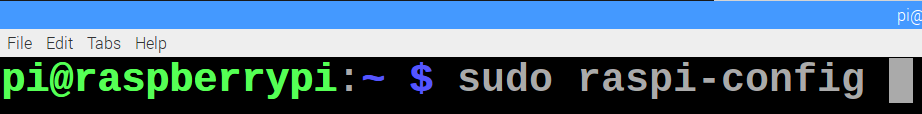
After Booting into the OS, setup the WiFi. If you are connected to the internet using Ethernet you may skip this step. Select the Wi-Fi button outlined in the image above.

[](https://raw.githubusercontent.com/V-Socrates/Is-It-Moist/master/Documentation/IMAGES-VIDEOS/Build%20Related/RPI2.png)

Select your connection from the dropdown list.



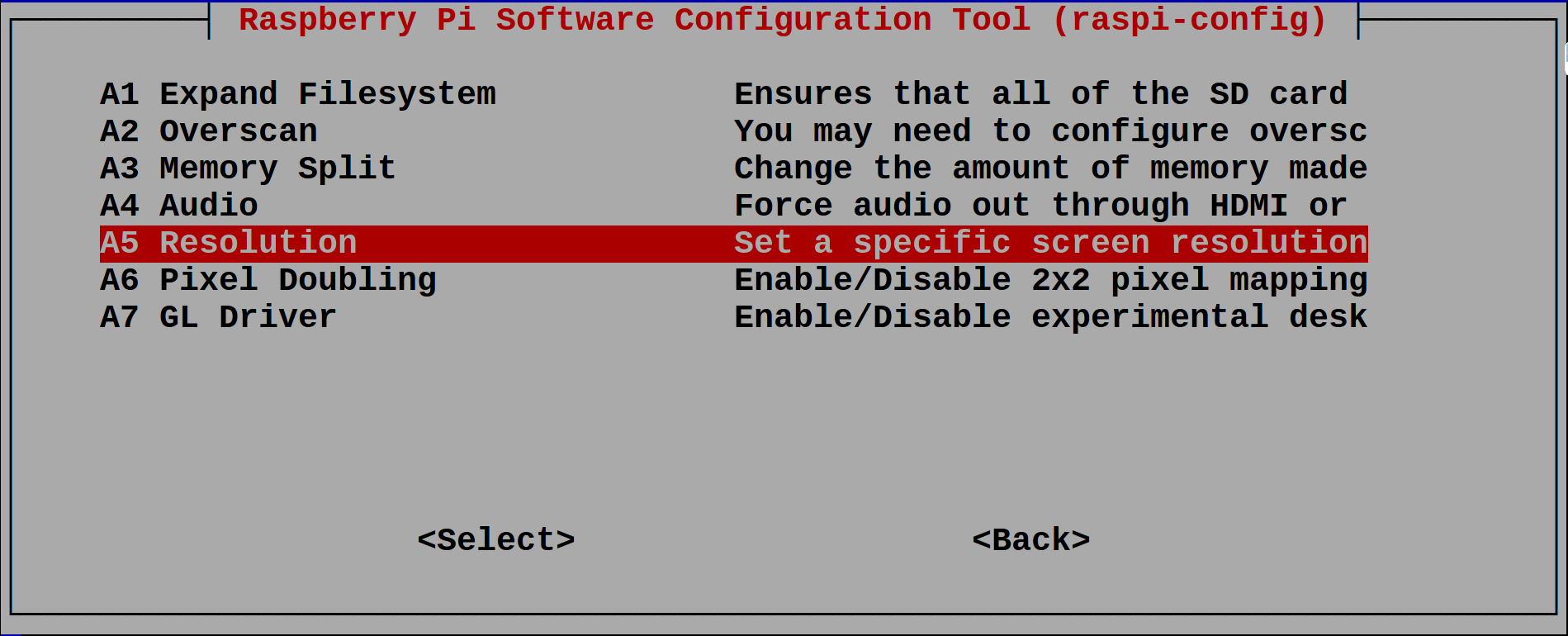
Enter you password and click OK.



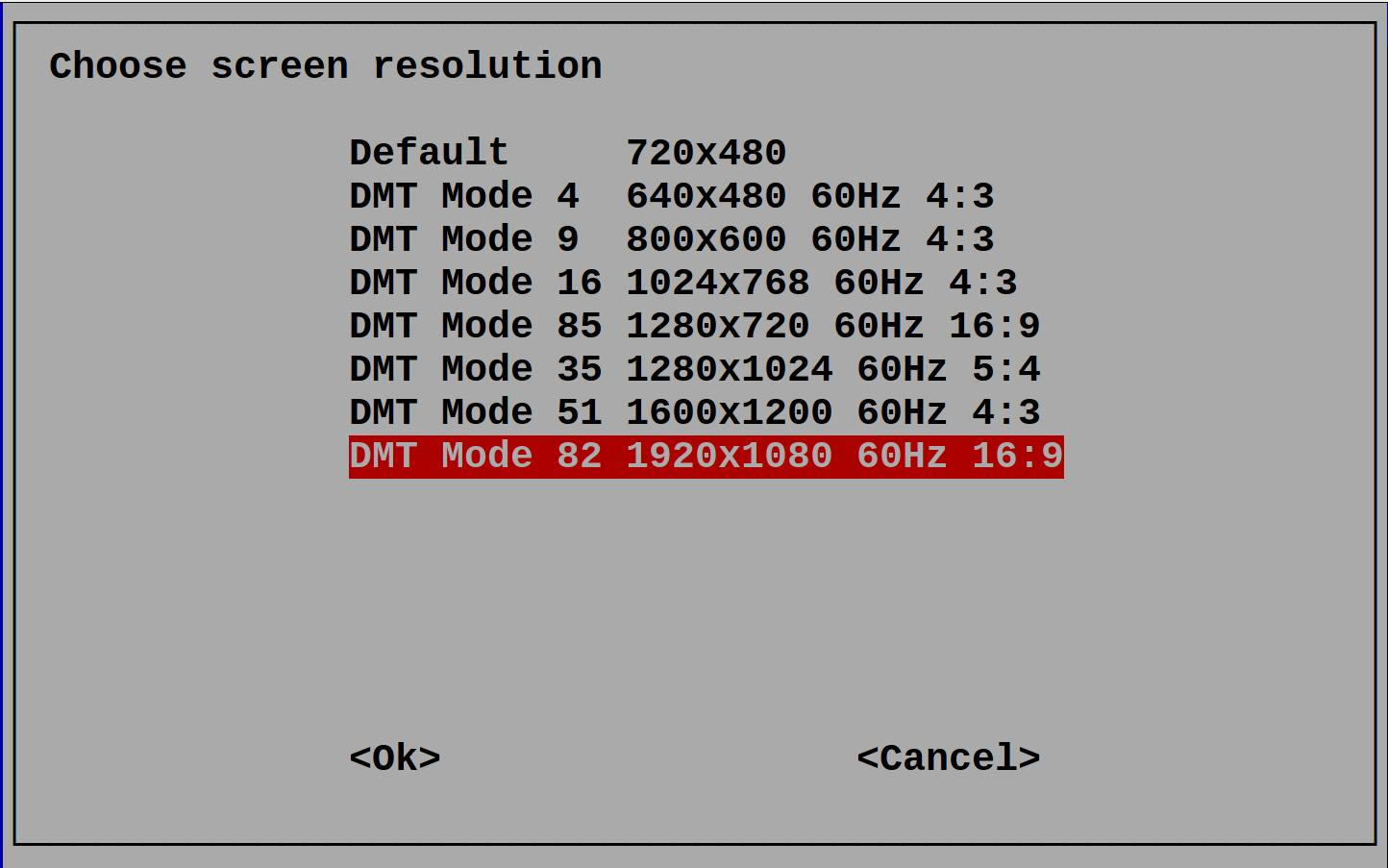
Open terminal and enter the command “sudo raspiconfig”.



Select Advanced Options.



Select Resolution.



Select your desired resolution. The default for this project is “DMT Mode 82 1920x1080 60Hz 16:9”. After selecting your resolution, restart your unit.

Enter the following command to setup the RaspiGuard unit.

Restart unit.

After restarting open terminal and run the following command.

If all aspects are configured properly, the unit should be running smoothly.

### 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

Dear Kristian,

Project RaspiGuard is currently on schedule and continuing with no significant difficulties. Project Is-It-Moist has been successfully integrated. The website design is complete and currently online. The hardware schematic design is completed and is currently being used to develop a custom PCB. Plans going forward include creating and testing PCB, Android application development, and hardware testing and scripting.

[**Progress Overview**](https://github.com/V-Socrates/RaspiGuard/raw/master/Documentation/Progress%20Overview.docx)**(**[**https://github.com/V-Socrates/RaspiGuard/raw/master/Documentation/Progress%20Overview.docx**](https://github.com/V-Socrates/RaspiGuard/raw/master/Documentation/Progress%20Overview.docx)**)**

[**Budget Overview**](https://github.com/V-Socrates/RaspiGuard/raw/master/Documentation/Budget%20Overview.docx)**(**[**https://github.com/V-Socrates/RaspiGuard/raw/master/Documentation/Budget%20Overview.docx**](https://github.com/V-Socrates/RaspiGuard/raw/master/Documentation/Budget%20Overview.docx)**)**

**Current Progress**

Vivek Socrates

Successfully integrated project Is-It-Moist into RaspiGuard. Hardware schematic has been created and is currently being tested. Currently writing and testing hardware scripts for accurate sensor readings.

Initial analog to digital converter MCP3008 and adafruit PiTFT incompatible for simultaneous use. Issue resolved by changing converter to ADS1115. Moisture sensor probe was tested and determined to be subpar. Probe has been replaced with superior product manufactured by SparkFun. adafruit PiTFT having compatibility issues on Rasbian Stretch. Issue is currently being investigated, possible resolution may involve using older Rasbian operating system.

Karel Tutsu

MySQL server setup and updated to support current hardware readings. Website template designed and created. Currently testing website and working to adapt previously created android application to match current hardware specifications.

MySQL database was initially designed to accommodate only two sensor readings. Tables and database have been updated to accommodate new hardware functionality. Website template needed multiple versions due to hardware functionality being undetermined. Issue is resolved due to new definitive hardware specifications. Previously written PHP scripts to access database are obsolete. New scripts currently in progress.

Heakeme Williams

After recently joining the team, is quickly becoming familiar with project details, and tasks. Undertaken the tasks of designing, creating and testing a custom PCB build based on hardware schematic.

Unfamiliarity with Fritzing software that was used in creating hardware schematic design. Issue being resolved by exploration of the software and online research. Conversion of hardware schematic to creating PCB schematic using EAGLE software proved problematic due to default component libraries lacking necessary components. Resolved by searching for custom libraries online. With updated functionality, PCB design needed to reflect new changes. Changes to PCB are currently in progress.

Regards,

Vivek Socrates

## 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

RaspiGuard is a monitoring system that is easy to setup and use. RaspiGuard offers multiple ways to track and monitor door and light activities, as well as the moisture level of plants. Users can access the information easily by navigating to our webpage or using our mobile android application. The alarm features on the mobile application keep user’s aware even while on go using notifications. The unit has a LCD touchscreen display showing live up-to-date readings with a built in alarm that can be set using the mobile application or webpage. A budget of $200.00 CAD was set which was not breached. The total cost to produce each unit is $---.-- CAD. This cost includes all components and shipping costs. RaspiGuard is a cost effective and simple solution to easily monitor any room.

# 5. Recommendations

<TO BE FILLED>

# 6. Technical References

Williams, M. B. (2017, September 09). Raspberry Pi Security System with Motion Detection / Camera. Retrieved February 03, 2018, from: https://www.hackster.io/FutureSharks/raspberry-pi-security-system-with-motiondetection-camera-bed172

Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

Dechuan Chen and Meifang Wang, "A home security Zigbee network for remote monitoring application," 2006 IET International Conference on Wireless, Mobile and Multimedia Networks, hangzhou, China, 2006, pp. 1-4.

doi: 10.1049/cp:20061246 keywords: {MSP430F135 micro-controller;Zigbee;remote alarming;security system for house;wireles network},

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5195407&isnumber=5195326

# 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>