Dryer Reminder

Michael Mohler

CST-451 Capstone Project Proposal

Grand Canyon University

Instructor: Mark Reha

Revision: 2

Date: 10/17/21

**ABSTRACT**

The Dryer Reminder is a device that can pair with a phone application to tell the user when their dryer has stopped spinning. Notifications are so ingrained into people that they wish they had them for the appliances in their home. The way dryers notify people on their own isn’t enough for many. The noise alert from a dryer can be missed after the jingle has finished. The timers are not consistent either to just match it with a phone timer. A notification from a phone would easily grab any user’s attention. Stay at home spouses, the hard of hearing, and tech enthusiasts would be ready to grab this for the convenience of not missing when their dryer has finished.

It might seem hard to develop this for any kind of dryer, but all the device needs to do is check if it is shaking. Every dryer needs to move to dry the laundry out, which makes it vibrate. With research to find out what component to check for motion, it just needs some programmers for the device and phone app to get it off the ground. There should not be a need for a dedicated server, so the cost after it is finished would go to upkeep on the mobile application. The device could be sold by itself for much less than a new dryer that has the same feature installed. There are huge communities for some of the small computers they use for this. Focusing on that would allow for selling the application by itself too. There are a lot of opportunities with the Dryer Reminder project with just as many payoffs.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 9/26/21 | Michael Mohler | Initial draft for review/discussion |
| 10/17/21 | Michael Mohler | Fixed some issues with the abstract, System Diagram and Schedule |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**Project Approval**

Professor Mark Reha

**TABLE OF CONTENTS**

Project Overview and Project Objectives 4

Project Scope 5

Project Success Measures 6

Project High-Level Solution 7

Project Controls 8

Project Cost and Schedule 10

Appendix A – References 11

Appendix B – Copyright Compliance 12

Project Overview and Project Objectives

**State the Problem and Background**

Many people live in a situation where their washer and dryer are out of way in their home. This can be a real inconvenience when using them as the only way for most users to get alerted that they stopped is by the noise they make when they are finished. If that noise is not heard, either through hearing difficulties or tunnel vision, then the user could be wasting their time and must run it again to prevent wrinkles. A notification to a phone when it happens would be great as most rarely miss it. Smart washers and dryers have these features built in but cost around $1,000, which is expensive for one feature. Being able to know immediately when one’s dryer is finished is convenient and could save time, but cost is a big barrier. It is less likely someone will need a washer that sends notifications too as the dryer is the bottleneck of doing laundry, especially if someone is running multiple loads.

Someone could just sync the timer on the dryer to a timer on their phone. It can be difficult to get a real time of when it will stop as modern dryers check for certain conditions before they are finished. Some even end prematurely, which can be annoying to find out half an hour later. As such the timer is only relative and does not give the user a true solution. It would also be helpful for a forgetful user to have a reminder somewhere that they have not picked up their laundry yet, instead of turning off a timer and forgetting about it.

There is one other way to see if a dryer has stopped too, motion. They are constantly tumbling the laundry to help with the drying process and as such the entire thing shakes. This is useless for a person though as someone would have to be close enough to the dryer to already tell that the machine has stopped. For a computer though, that motion can be the best way to decide when the dryer is obviously off and on. Overall, a cheap solution for users to get a notification is something that would intrigue anyone who is forgetful of their laundry.

**Project Objectives**

* No major bugs that ruin the user’s experience.
* Response UI for the Android App
* An outlook on further development in apps or embedded systems.

**Challenges**

* The team has not posted between the Pi and another device.
* The team is inexperience with Android App Development
* There may need to be proprietary software made for the hardware.

**Benefits and Opportunities**

This project will be a good step forward with experience in many fields. Python, mobile application development, embedded systems, and more. Python is an ever-growing language that is the third most popular language currently. Experience with python in embedded systems is a real-world use of the language. Having that system communicate through a phone app is a common item one would see sold on the market today. As such it would be a useful skill to have experience in that communication. Mobile applications are a multibillion-dollar industry with ANAS stating, “The final quarter of 2019 showcased this potential with an estimated consumer expenditure of $21.9 Billion on apps across Appstore and Google Play store” (ANAS, 2021). There is a lot of potential here to learn a lot about the industry and take our creative mindset even further.

Project Scope

**In Scope:**

* Accelerometer
* Pi
* Python
* Android Application
* Phone Notifications
* Exception logging
* False positive prevention.
* Mobile Application Controlled

**Out of Scope:**

* Develop proprietary software
* Adjust shake detection
* Washer detection mode

Project Success Measures

After making the project open source, it would be nice to see how many compliments or star it on GitHub, after posting it in a pi group. This would show how many would have been interested in purchasing the product, proving my skills and its function. During the capstone project showcase, it would be great to see the project get above average interest relative to the other projects. If anyone were to want to hire me at any point based on this project would be a sign of its overwhelming success for me.

|  |
| --- |
| Project Completion Criteria |
| 1 – 20 People compliment the project after it is complete outside of the school. |
| 2 – 5 Scouters have interest in the project’s development. |
| 3 – 1 person interested in hiring me based of the project’s features. |

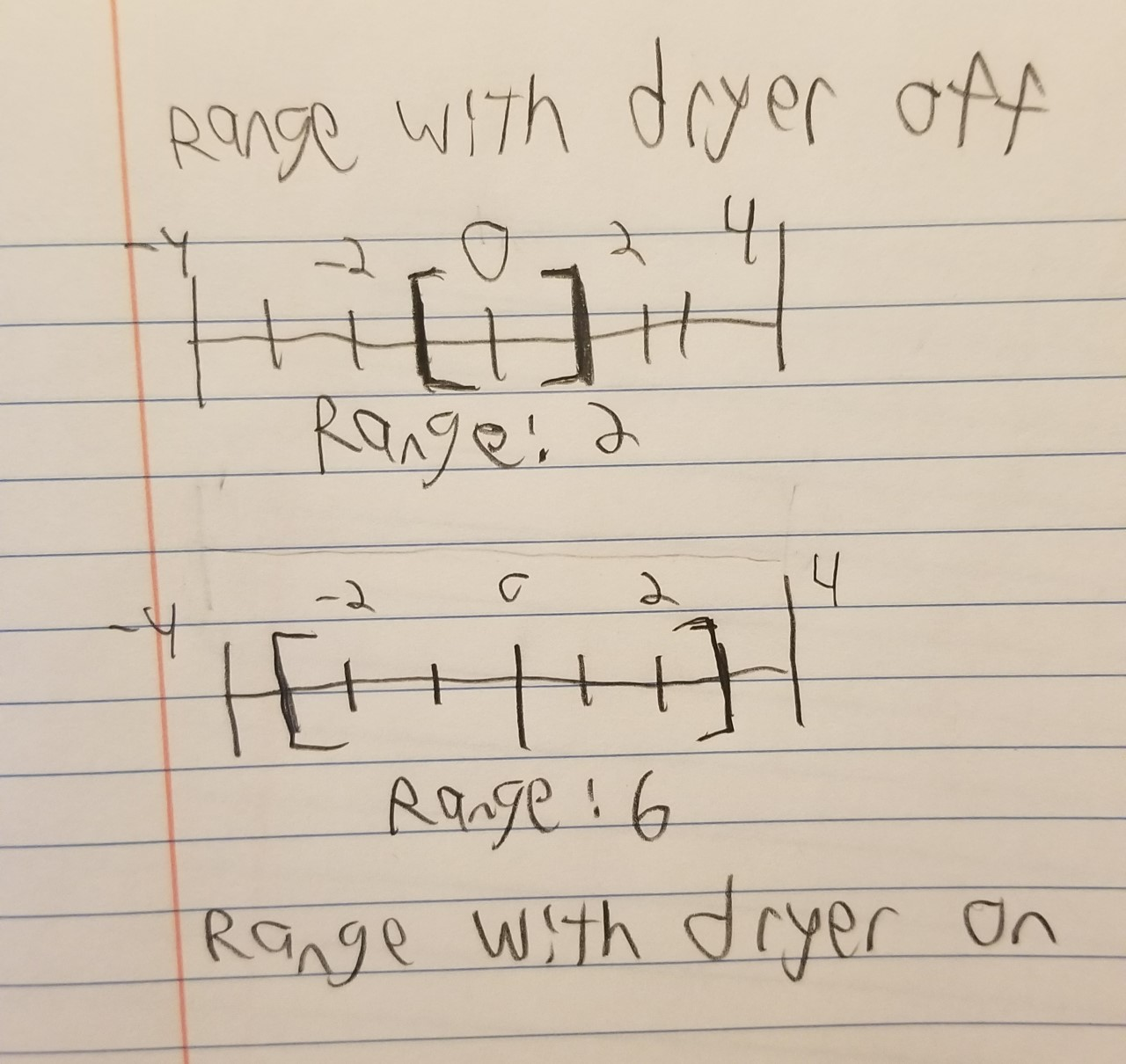
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | Every Dryer shakes enough for the Pi to detect it. | It is impossible to check every dryer’s vibration. | Constraint | Good | 09/16/21 |
| 2 | Dryers run longer than washers do | If a user’s dryer finishes regularly before their washer does, it might feel like the Dryer Reminder does not help with the flow of their laundry. | Assumption | Good | 09/20/21 |
| 3 | The Pi doesn’t change its initial position when the dryer shakes. | The dryer shaking could move the Pi too much and possible give the device inaccurate results. | Assumption | Good | 09/26/21 |

Project High-Level Solution

**Introduction**

Everybody has forgotten they were doing laundry before and have left it in their dryer. Some people actively do it, but others get distracted. Dryers might have all the tools on them to notify people when they are done, but it’s easy to get caught in the moment and miss them. After that jingle they make it’s not easy to just notice that there is no noise coming from the laundry room. That is why for some, it would be better if their dryer sent them a notification to their phone when it’s done. It’s hard to miss that no matter what they are doing. This is how the Dryer Reminder came into conception.

    The two ways that people can usually tell when a dryer is done is by the noise and the timer on the front. The jingle modern dryers make can easily be missed though and who knows what kind of tunnel vision someone will have ignored it. Timers on dryers are not relative to actual time though as there are other factors involved that change the time. However, there is another method to tell how a dryer stops running and its movement. The shaking a dryer makes could be used to tell when it has stopped. Accelerometers are so cheap and reliable now it could be possible to detect the shaking motion on a device that could reliably send a notification to the user. This does bring up the assumption that every dryer shakes enough for an accelerometer to read the difference. Accelerometers detect movement in the x, y, z axis, going along with positive and negative values to show which way in that axis it is going. They are usually used to detect their current position, but they can also be used to detect where they were instead. There is no need to get into vectors or calculus either as each axes range will be calculated independent from the other.

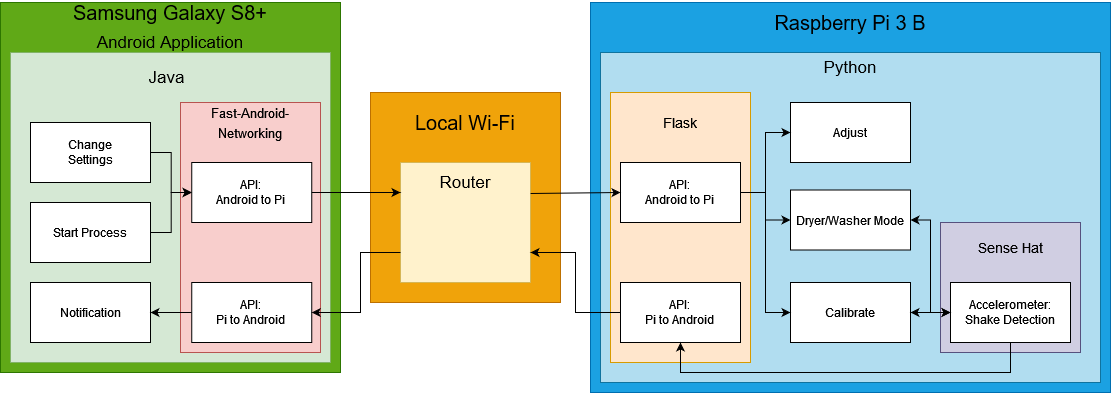


    This project will be built for dryers and not washers as the dryer is more of the bottleneck when doing laundry. Knowing when a washing machine stops does not help if the user’s dryer is in use. There will be an attempt to get the device working with washers, but as of now that feature is out of scope. Washers have moments where they do not move for extended periods of times, and as such will be hard to detect when they stop based on movement alone.

**Solution**

The Dryer Reminder works as such. A Raspberry Pi with an accelerometer will be attached to the dryer. An Android app will be connected to the Pi to control it and alert the user. The user will first use the app to get the settings of the dryer being off before starting it. Through local Wi-Fi, an API will be used to communicate between the Pi and App. After the dryer is running the user will user their app and it will alert the Pi to start checking for if the dryer has stopped. Using an accelerometer, a difference in movement will be used to detect if the dryer is no longer shaking. An API over the local network will POST to the user’s phone app that the dryer has stopped. It will then send a notification to the user to tell them their laundry is dry, and they can grab it. The user will confirm on their app that they got their laundry, and the device is ready for use again.

**System Block Diagram:**



An accelerometer attached to the dryer should be able to collect constant data on how much it has moved in each axis. Accelerometers will detect very little amount of movement, even if there is not any. So, it will not be as easy as sending an alert once the motion value is zero. The best way to determine the difference between two sets of motion would be to calculate the range. Range is calculated by taking the largest value in a data set and subtracting it by the smallest. In theory, the larger the range the more movement there is. That does mean that a data set will need to be saved before the dryer starts to get an idea of what the range is like when the dryer is not moving. Here is a simple math equation to get an idea of how it can be calculated.

**Saved Range = (Largest value - Smallest Value) \*When dryer is off**

**Current Range = (Largest value - Smallest Value) \*When dryer is on**

**Saved Range < = Current Range = Dryer has stopped**

This will need to be calculated for each axis individually. It is likely there will be moments that the accelerometer will record data in such a way that will make the range mimic there being no movement when there is. This would result in a false positive and can be annoying to the user. Checking all three axes though gives the chance of mitigating that. If the alert that the dryer is done is only sent if two or three of the axes meet the requirements in that set, then it is unlikely to trigger a false positive. Even more could be done as prevention, like increasing the number of times the accelerometer records data in each set or only sending the alert after multiple positive results in succession.

The project will use local Wi-Fi to communicate between devices as most people will be in their own home anyway to do their laundry. This will prevent the need of a server to handle the API’s which will save on cost and avoid any issues involving upkeep or fears of servers going down and preventing the application from working. An Android app is only being developed since it is the only phone the team uses. If the project is shown to have great promise, then apps for other devices will be considered as well.

The Dryer Reminder has a great chance at making people never forget their laundry again. Mitigating getting sidetracked by other things and being forced to run it again, repeating the cycle. By the end of this project the people who have their hands on this will be happier and our team will be glad to have taken the initiative to learn so many new things that make up this industry. Knowledge on response UI, embedded systems, and proper debugging and testing to make the project run perfectly. We have a lot to look forward to.

Project Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| Trouble Making app | High | The entire app portion of the project must be changed. | Study up on android app development | Move Pi Notification to computer instead of app. |
| Posting between App and Pi | Medium | Notifications may need to be delivered by another method. | Practice between posting between Pi and a computer | Work with the Pi’s Bluetooth instead. |
| The Pi or accelerometer breaking | Low | Slowing down the project development, especially in testing phases. | Makes sure to properly protect and secure the device while in use. | Have multiple Pi’s and accelerometers when the final versions are decided. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | Accelerometer Library might not be Open Source | Work will need to be delegated to find or develop for the accelerometer. | One will have to either find an Open-Source Option or Develop our own for | Michael | High | 9/16/2021 |  |  |
| 2 | Time Management with other projects | Reduces time to study, practice, and research solutions for the challenges. | Delegating time using proper time management software. | Michael | Low | 9/20/21 |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 | Change Android to Computer Program | Low | Michael | 9/24/21 |  |  |  |  |  |
| 2 | Use different accelerometer | High | Michael | 9/24/21 |  |  |  |  |  |

Project Cost and Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Initial Cost | Quantity | Total |
| MPU6050 | $3 | 3 | $9 |
| ADXL345 | $8 | 1 | $8 |
| Sense Hat | $42 | 1 | $42 |
| Pi 3+ | $35 | 2 | $70 |
| Pi 3 Sense Hat Case | $7 | 1 | $7 |
| SD Card 32GB | $7 | 1 | $7 |
| Pi Power Supply | $10 | 2 | $20 |
| Strip Magnet | $1 | 10 | $10 |
| Google Developer Account | $25 | 1 | $25 |
| Total |  |  | $198 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Task | Dependencies | Status | Effort Hours | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Find Accelerometers for Pi | Accelerometer libraries | Complete | 3 | 9/10/21 | 9/11/21 | 0 | 9/15/21 | Pi Site |
| 2 | Accelerometer can detect dryer shaking |  | Complete | 5 | 9/16/21 | 10/03/21 | 0 | 9/23/21 | Pi, Accelerometer |
| 3 | Print to the console when the dryer is and isn’t running. | Python time library | Complete | 3 | 10/3/21 | 10/17/21 | 0 | 10/10/21 | Pi, Accelerometer |
| 4 | Pi Sends Email when dryer stops | Smtplib, ssl library | Removed | 5 | 10/17/21 | 10/31/21 | 0 |  | Pi, Accelerometer |
| 5 | Posting with Java on computer | Java API | None | 7 | 10/31/21 | 11/14/21 | 0 | 10/13/21 | Computer |
| 6 | Make a Simple Android Application | Java, Android Studio | Complete | 5 | 11/14/21 | 11/28/21 | 0 | 10/26/21 | Computer |
| 7 | Post between Java App and Pi |  | Complete | 3 | 11/28/21 | 12/5/21 |  | 10/18/21 | Computer, Pi |
| 8 | Develop App that sends notification when simple condition is met. |  | None | 15 | 12/5/21 | 12/19/21 |  |  | Android Phone, Android Studio |
| 9 | Notification sends when app isn’t running. |  | None | 3 | 1/3/22 | 1/16/22 |  |  | Android Phone, Android Studio |
| 10 | Pi communicates with the App when simple condition is met |  | None | 6 | 1/16/22 | 1/30/22 |  |  | Android Phone, Android Studio, Pi, Accelerometer |
| 11 | Pi Communicates with app when dryer stops shaking |  | None | 5 | 1/30/22 | 2/13/22 |  |  | Android Phone, Android Studio, Pi, Accelerometer |
| 12 | Change  Shake Detection levels |  | None | 3 | 2/13/22 | 2/27/22 |  |  | Android Phone, Android Studio, Pi, Accelerometer |
| 13 | Change settings with phone |  | None | 6 | 2/27/22 | 3/14/22 |  |  | Android Phone, Android Studio, Pi, Accelerometer |
| 14 | Washer Mode |  | None | 10 | 3/14/22 | 3/28/22 |  |  | Android Phone, Android Studio, Pi, Accelerometer |

Appendix A – References

ANAS, SHAH. (2021, April 21). How much money can an app make in 2021. TekRevol. Retrieved September 26, 2021, from https://www.tekrevol.com/blogs/how-much-money-can-app-make/.