Student Name : Cian Dunne

Project Repo URL : https://github.com/CianFD/CompSysAssignment2/tree/main

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grade Band** | **Combined Knowledge** | **Networking Technologies** | IoT Solution | **Communication** |
| **Base** | I believe I have achieved this section | I believe I have achieved this section | I believe I have achieved this section | I believe I have achieved this section |
| **Good** | I believe I have achieved this section | I believe I have achieved this section | I believe I have achieved this section | I believe I have achieved this section |
| **Excellent** | I have used multiple concepts from programming (creating my scripts, incorporating loops, importing libraries, calling other scripts from inside methods), webdev/ict skills (creating a glitch web app), database (creating realtime firebase database and storage), and computer science (connecting physical hardware of sensor, buzzer, and RPi Camera to RPi using Ribbon Cable, T-Shaped Connector and Breadboard) | Once data has been acquired by the RPi through physical wires and voltage, it is delivered to Firebase Storage and RealTime Database over Wi-Fi. This Data is passed from Firebase to a Glitch Web App to view. Furthermore through ThingTweet a command is taken at the time of execution which delivers a tweet with a URL link to the Web App to a Twitter feed that the Owner is subscribed to. | I believe this to be of good prototypical standard - the device created represents a useful machine for both teaching dogs not to bark unnecessarily and alerting their owner to the occurrence of the barking and informing them what they’re barking at (be it something irrelevant like a passing car or something important e.g. an intruder trying to enter). Production System is simplistic but has multiple lines of communication making for ease of build/use. I believe the solution to be novel in that it’s not a product currently released (to my knowledge) but it could be a very useful product for dog owners (especially those training puppies - speaking from personal experience) | I have created multiple pieces of documentation detailing the work I have created - an initial proposal was submitted, followed by Benchmark Graphics which displayed an Informational Flow Diagram, a Block Level Pictorial Diagram and a Logic Algorithm Flowchart. At time of submission a video walking through the demo of the project and breaking down the code used was submitted, along with a list of external references used, a document displaying the physical components and how they connect and a short demo video explaining to a customer what they would have if this product were real. All of this content was uploaded to a Git Repository. A readme file was also included giving a basic breakdown of the machine and how everything interacts. |
| **Outstanding** | I have used a number of outside references in conjunction with the knowledge available in the course - see my references document | I would like to say I have reached this section but I am uncertain - see above | I am unsure if I have reached this section - see above (I’m inclined to say no I didn’t get to this point) | A basic Glitch WebApp was created displaying the latest Photo from the RPi Camera Module pulled from the RealtimeDB Firebase. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Additional Comments:

Website Links:

(not sure if you have access to my firebase storage but including links anyway)

<https://console.firebase.google.com/project/dogbarkalertdatabase/overview>

<https://console.firebase.google.com/project/dogbarkalertdatabase/database/dogbarkalertdatabase-default-rtdb/data>

<https://console.firebase.google.com/project/dogbarkalertdatabase/storage/dogbarkalertdatabase.appspot.com/files>

<https://glitch.com/edit/#!/dogbarkingalertweb?path=README.md%3A1%3A0>

<https://dogbarkingalertweb.glitch.me>

<https://twitter.com/CDWIT1>

Bugs:

1. The Sound Detection Sensor appears to go into an unintended Feedback loop where the initial noise is picked up by the Sensor triggering the Buzzer but the noise the Buzzer makes is then triggering the Sensor when the Loop reoccurs so the only way to stop the Buzzer ringing is to CTRL+C the script. Attempts were made to alleviate this but a fix was not found.
2. In earlier versions of the script new frames created would be given names frame 2 frame 3 and so on. To tidy this up I deleted new frames from the RPi, and Firebase Storage and Firebase realtime database. When incorporating firebase and thing tweet elements all of the frames taken became labelled as frame 1 and despite multiple loops (as mentioned above) the camera was only taking a single image per script ran - as only one image was taken the frame name became always frame 1 so it was overwriting the previous image when a new one was created. This did not affect Firebase Realtime Database as it could differentiate new files based on the timestamp but Firebase Storage was overwriting each file as it arrived.

