

Greetings, my name is Siddharth Bhardwaj and in this technical demonstration video i would like to present the final release of my system health reporting application in this video i would like to demonstrate transition state beta where my raspberry pi and base station pc have already booted up and as such the pi will send data about its health to the base station continuously while the base station will listen and receive the data from the pi about itself continuously then i'm going to demonstrate transition state omega where i tell my pi to reboot while my base station continues to listen and as such upon reboot the pi will automatically send the data about its health to the base station along with these two transition states i would also like to demonstrate a error push notification feature where i inject fault into my application through the means of a config file and then demonstrate how the application sends a notification to the user who is monitoring the health of the microcontroller.

Now i would like to talk about the demonstration setup so on the left here i have the raspberry pi into which i've ssh into which is in ambulance a on the right here i have the base station pc which will listen to the data that the raspberry pi will transmit about itself and to the right of that i have the database in which the the system health data that gets transmitted by the pi will get stored and what you will see here as as the data comes in into the spreadsheet it should update the visuals automatically and what's key here is the updating of the visuals on the website which is a dashboard about the system health system health of ambulance A.

So the url for the website is available on the documentation site but i have a bookmarked here so i'm going to log in as the base station personnel.

And what i'm concerned about in this application is the health of ambulance A.

A quick aside the visuals here on this website are generated using javascript where on a refresh i've programmed my html page refresh after a certain period of time after which it executes javascript client side and reads the spreadsheet and upon reading the spreadsheet it creates this visual real time and if you were following my technical demonstrations um videos previously you would notice that i've done a test about reliability of data propagation with respect to taking simply this embedded object this visual and then putting it onto a website versus using javascript to generate a graph real time using google chart api.

So that's enough about the setup what i would like to show next is the gyroscope sensor the gyroscope sensor is important because it simulates the operation of a bvm a bag valve mask automated ventilator that would be operating inside of the ambulance in essence in my code of the raspberry pi i have set x y and z threshold that model the angular velocity in the respective axes and as such i have used a config file with json format to set threshold values in these specific accesses and then program my python code to either transmit a okay or an error based on the thresholds that i set i will deep dive i will dive deeper into my code in a later section of the documentation which you will find on the team website however in this case i just want to demonstrate that the gyroscope is connected to the pi and is operational at the moment.

So as you see here the raspberry pi is well sorry the gyroscope sensor is continuing to transmit data about the x-axis the y-axis and then the acceleration in respect respective axes along with the rotation value here's the code for it as i said i will i dive deep dive

deeper into the code in a later section of the website in a different video but in short i've written a python script that reads the data from the sensor which would which would be connected to a bvm unit and then transmit the health of the system which is connected to the mic the microcontroller which would then transmit the data about itself.

So now that that's aside i would like to jump into the demonstration of transition state beta.

How i would like to show this first is by telling my basestation pc to listen.

Now that that's listening I'm going to then adjust my windows before I start sending the data from the raspberry pi inside the ambulance so I want to first just demonstrate these two and the data coming in so what you should see is the data being populated here.

And so as you see as data is sent from the pi to the base station it is received and upon being received it is then stored into a database onto the cloud which then updates the visuals as you see here automatically so just one more time it's going to wait for one more transmission yep perfect you see three there and then you see this also update though again what's key is the update of the graphic on the website so after 30 seconds this website will update and what you should see is the same amount of the same visual as you see there.

Perfect, so that's updated and as you see here the last transmission was on 200 perfect and you see the last transmission's last transmission on 2800 for ambulance a on the website that's publicly available.

So now that now that the demonstration for transition state beta is finished i would like to demonstrate transition state omega what i'm going to do is i'm going to tell my pi to stop transmitting data about its health and then i'm going to ask it to reboot.

And what you should see is upon reboot successful successful reboot the data will start to propagate here and my base station is continuing to listen so upon reboot it will then receive the data and then populate it in the web base into the database which would then subsequently update the data being read by this by this dashboard so i'm going to pause the video for a few seconds all right so as you see here the um the data is starting to come in so i left my pi as it was i my base station pc is continuing to listen and as you see here the records are starting to come in it's going to show one more iteration perfect and what that means is now the website should be updated as well i'm just going to pull it up.

So the last transmission was here.

Now i'm going to wait for it just for a few more seconds to update perfect and you can see the updated visual on here as well.

So that wraps up the demonstration for transition state omega now i want to jump into the demonstration of the push error notification feature so what i'm going to do is as the data is continuing to come in which you see here on the right you can see that it will also be received here as in okay what i'm going to do is i'm going to ssh back into my pi and inject fault using the config file i showed earlier.

And now that i'm in i'm going to open up the config file.

And i'm going to then change the value of this x threshold value.

And now what you should see is you should see error popping in.

And as errors is received you should see a notification being sent to my email which says what the error code is the date and time for the error code along with the ambulance id so that's why i received two error notifications sorry two errors and as you see i've received two notifications now i do want to call out since i'm using amazon sns it requires a subscription in this case i'm being received um i'm i'm being sent the uh notification i could confirm it and that would just uh take away this email however regardless of that i still get continued to i still get the the notification being pushed.

Though i dive deeper into the code in a specific in a different section i do want to show um a bit about how the error gets sent and and what it's being sent uh so um i am sending a error notification that has the um as i said the message the time and date for it and the ambulance id and in this case use what you saw on the screen was an email that gets sent to me via amazon sns i actually have um a notification being sent to a cell phone specifically my cell phone um though i'm unable to stream the uh the screen of my cell phone um real time at the moment and so this is how the data this is how the error notification gets sent programmatically through my code now what i would like to demonstrate also is how do i how i'm going to fix the error the key thing to note here is the application is still running the the records are still coming in as you see here and and on the website it's updating the website also as you see here and so if i want to fix this error i'm not going to stop my application i'm simply going to just adjust my file a config file which is going to from which the changes are going to reflect in this application during one time.

So going back into the application and adjusting the negative to a well added negative back and so what you should see now is you should see okay coming in.

Yep, so you see the okay come in and then confirm this on the website.

Just gonna wait for it to update.

Perfect so as you see here the most recent one was 20 hundred hours and eight seconds and you see the okay come in on the and you see the changes being reflected on the web application as well.

In conclusion i demonstrated transition state beta transition state omega and the error push notification feature for this application i showed you how i was sent data from my raspberry pi and received data on my base station which continues to listen and then how the data gets stored inside of the database and then how from the database using javascript i update or create a chart a visual analytics visual on a website and this website is publicly available and it can support multiple ambulances in this case we were working with ambulance 8 but it can also support ambulance b and that concludes the demonstration for this final release thank you for your time.