IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 09, Issue 7 (July. 2019), ||S (II) || PP 32-38

Smart Glasses for Blind People Based on ARM Embedded System Module

Dr. P. G. Kuppusamy[1], Y. Murali[1], M.Afsar Ali[1], P. Pavan Kumar[1]
[1]*Department of Electronics and Communication Engineering, Siddharth Institute of Engineering & Technology (SIETK), Puttur, Andra Pradesh, India*
Corresponding Author: Dr. P. G. Kuppusamy

Abstract: Ever thought how the life of a blind person, their life is full of risk. Theycan't even walk alone through a busy street or through a park. They shall needsome assistance from others. They are also curious about the beauty of theworld; they should have will be the excitement to explore the world, and to beaware of what is happing in front of them. Even though they can find their ownthings without anyone's need. So, there is an idea called SMART GLASSES.Simply, smart glass is a pair of glasses for the blind. By using smart glasses, aperson can able to know what is going on in front of him/her. Smart glasses isdeveloped with raspberry pi 3 and google android things. When the personpress the button on the smart glasses, the camera will take a picture andanalyse the image using tenser flow and detect what is that picture is about thenusing speaker or headphone, the google android will voice assist the personabout the picture.

Keywords: ARM Embedded System,

Date of Submission: 11-07-2019 Date of acceptance: 26-07-2019

I. INTRODUCTION

Ever thought how the life of a blind person, their life is full of risk. They can't even walk alone through a busy street or through a park. They shall need some assistance from others. They are also curious about the beauty of the world; they should have will be the excitement to explore the world, and to be aware of what is happening in front of them. Even though they can find their own things without anyone's need. So, how we solve this? We are introducing Sight! Simply, Sight is a pair of smart glasses for the blind. By using Sight, a person can able to know what is going on in front of him. Sight is developed with a raspberry pi 3 and Google android things. Sight does have three main parts, a raspberry pi 3 (powered by android things), camera and a button. When the person press the button on the Sight, The Sight will take a picture and analyze the image using tensor flow and detect what is that picture is about, then using a speaker or headphone, the Sight will voice assist the person about that picture. This is how sight works!

Blind glass is an innovative glass designed for visually disabled people for improved navigation. We here propose an advanced blind glass that allows visually challenged people to navigate with ease using advanced technology. The blind glass is integrated with ultrasonic sensor along with light and water sensing. Our proposed project first uses ultrasonic sensors to detect obstacles ahead using ultrasonic waves. On sensing obstacles the sensor passes this data to the microcontroller.

The microcontroller then processes this data and calculates if the obstacle is close enough. If the obstacle is not that close the circuit does nothing. If the obstacle is close the microcontroller sends a signal to sound a buzzer. It also detects and sounds a different buzzer if it detects water and alerts the blind. One more feature is that it allows the blind to detect if there is light or darkness in the room.

II. IMPLEMENTATION OF HARDWARE & SOFTWARE

Implementation of hardware:

a) Create an OS for raspberrypi: Step 1:

The Android Things Console provides easy and secure deployment of updates to your connected devices. Google provides the infrastructure to host and deliver system and app updates with the developer in final control.

Download and install the latest Android Things system image. Build factory images that contain OEM applications along with the system image. Push over-the-air (OTA) seamless updates, including OEM applications and the system image, to devices manage and share OEM applications across products and owners. Monitor informative analytics to understand how well products are performing.

To get started, create a new product. Gather the following information:

- **Google account**—A Google account to associate with the product
- **Product name**—The internal name you want to use to refer to your product. We recommend that this name be different from the final marketing name. Developers can see this name in the console, but consumers don't see it. Maximum characters: 70
- **System on Module (SOM) type**—Hardware platform on which you are building your Android Things product
- **Product description**—A brief paragraph that provides more detail to identify the product and its function. Maximum characters: 100

Step 2:

Open the Android Things Console. If the **Welcome** page appears, sign in, agree to the terms of service, and click **Continue**. Sign in with the Google account that you want to associate with your Android Things product. The Android Things Product page appears. It lists the Android Things products associated with this Google account, if any. For new users, the Android Things Product page is empty. Click **CREATE A PRODUCT.**

The **Create new product** dialog appears.Type the product name for your product.This is an internal name that users won't see. Collaborating developers can use this name before deciding on the actual product name. You can change the name at any time after project creation.

1. Select your hardware platform from the **SOM type**list.
2. Type a description of your product.
3. Click **CREATE.**

Step 3:

The product page in the Android Things Console allows you to change the product settings, create different models of your product, and share products with other Android Things Console users.If you are not already on this page, open the Android Things Console and click a product you previously created.

With the Android Things Console, you can create multiple models of the same product. You can use models to create variants of application builds that all run on the same underlying hardware.When you create a new product, the Android Things Console automatically creates a unique model identifier under **Models**. Click the ellipse next to this ID to change it. To add another model, click **ADD A MODEL**.

Under **Product settings**, you can edit the product name or product description, if desired. If you do, make sure to click **SAVE CHANGES**.To share your product with other Android Things Console users, click **ADD MOREPEOPLE** under **Sharing**.To share your product with a Google Group, enter the group email address. Members of the group can then access your product according to the access rights you set.If you are a member of a group and would like to remove a shared product from your list, ask the group administrator to remove you from the group.

Step 4:

The **BUILD** tab in the Android Things Console allows you to create builds for a given model. You can customize each build with a version of the OS and a set of applications; build resources, and hardware configurations. You can see the build history for a model on this tab as well.If you are not already on this tab, open the AndroidThingsConsole, click a product model you previously configured, and click the **BUILD** tab.

Step 5:

To get started, click the **NEW** button, select **Startfromscratch**, and follow these steps:

Name the build configuration

Select an OS version for Android Things

Select the applications you want to include in the build. Click **SELECT APPS**.

There are two types of applications: developer apps and Google apps. Developer apps are applications that you create and upload into your library. Google apps, such as Google Play Services, are created and managed by Google; the available versions are dependent on the Android Things OS version selected in the previous step. Both types of applications are compiled and packaged into a single file with the .apkextension (an Android application package file).To upload your own developer apps, click **UPLOAD NEW APP** on the **Include APKs from library** dialog and select an apk from your local drive. Make sure to select the check box next to the package name for the uploaded app.

Android Things expects only one application to expose a home activity for the system to automatically launch on boot. The Android Things Console will not generate a build that contains more than one app with a home activity.

Step 6:
Click the **APPLY** button.
Before you can add an app to the build configuration, you need to explicitly grant dangerous permissions

Step 7:

To add additional build resources, click **UPLOAD**. All build resources are undersigned and cannot be modified for a particular build after the build has been created.Google provides a number of built-in resources, such as the font package for Android apps.

Hardware configurations allow you to define peripherals and configure their attributes. To add a hardware configuration, click **ADD NEW PERIPHERAL**.**Note:** If you know the build will have a complex hardware configuration, you may want to select **NEW > Use Starter Kit settings** when you first create the build configuration. This will automatically add a set of peripherals for you to customize.

Step 8:

Edit the user partition size. This text box is found under the App Partition bar graph. The partition must be large enough to fit all of the apps, build resources, and hardware configurations specified in the earlier steps.

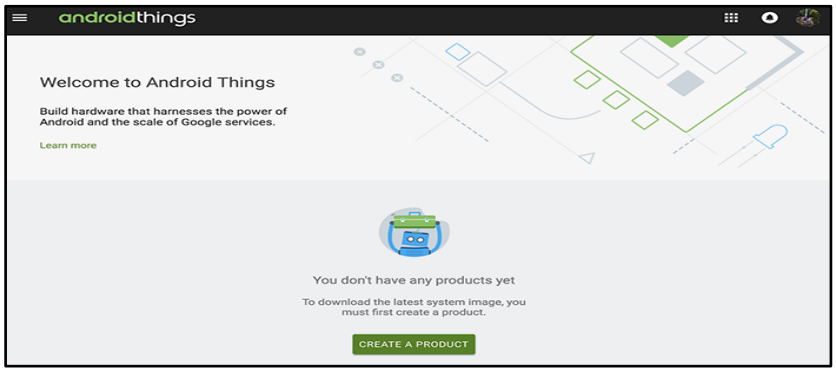

Figure.1 Console window

Figure.2 New create product window

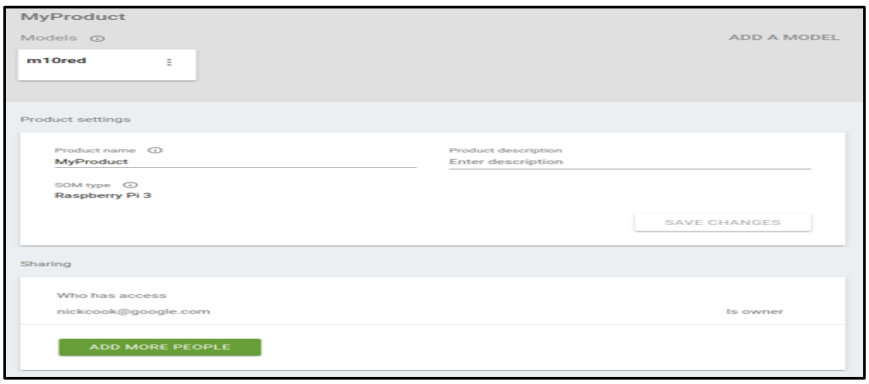

Figure.3 Create a models & product settings

Step 9:
Click **CREATE BUILD** to create a build using this configuration. When the build is complete, a new entry will appear in the **Build configuration list** table.
To view, copy, or delete a build, click the ellipses at the end of a build entry in the **Build configuration list** table.
Click the **Download** menu and select a version of the build to download:
- **Production**: Use this build when deploying for release. Production builds do not allow access to debugging tools, such as adb.
- **Development**: Use this build for development and testing. Development builds have additional debugging capabilities enabled.

Step 10:
A zip file will start to download, Wait until the download is complete.After download is completed. Unzip the file using **7zip** or **winrar**. Wait until the extraction is complete (probably it should take only 1 to 2 minutes).
After extracting the file you will get a **.img** file(this is the android thing OS for your Raspberry Pi)
b) Burning Android Things To SD card (Flashing):
Now you need to burn this img file into your micro SD card of the raspberry pi
We are using Etcher (Etcher is an open source project by resin.io).
Step 1:
Connect **SD card** to **Computer** via a **SD card Reader** and open **Etcher**.
Step 2:
Select **image** and select **SD Card** and hit **Flash**.
Wait for it. After the **Flash**, unplug the **SD Card Reader** and insert it into our **Raspberry Pi**
Now here our **Local IP** is **192.168.43.34.**
***c)** Setup Android Studio:*
Step 1:
First Download Android studio (Stable).After downloading, install and open **Android Studio**.
Step 2:
Open an existing project (SIGHT) by clicking **Open an existing Android Studio Project**
Step 3:
Open our project file.
Step 4:
Click OK to Open the Project. **Run the SIGHT on your Raspberry Pi**. But we need a connection between the **Android Things** device **(Raspberry Pi)** with our Android Studio for Upload and Debug our programs. We have the **ADB** tool (*Android Debug Bridge*).
It will act as a bridge between our **Android machine** (phone/things) and **development platform** (Android Studio) and help us Upload and Debug our program. So next we need to set **ADB** in between **Raspberry Pi** *(running on Android*) and **Android Studio**.
In Windows, open Android =>SDK=>Platform-tools=>adb
You can see the **adb.exe** file here, but we can't directly access it. So opena **command prompt** or **Power Shell** and get in the folder and just type code on **command prompt** or **Power Shell**.
We got the Raspberry Pi IP from the first step. And you'll get a response after the connection is established.
Step 5:
Now just upload the program by clicking the green triangular button.

Step 6:
Hit **Run**. You can see the **Raspberry Pi** on the listed of devices. Just click **OK** for running our program.

Software implementation:
Time for wiring up the circuit! Here we are using a button for Triggering the Camera to capture the image.Don't forget to connect Raspberry Pi Camera module and Earphone to the raspberry Pi 3
We build our own Pi hat on a pref board, we recommend you to make one yourself.
Connect the raspberrypi camera to the raspberrypi 3 module port.

a) Making Enclosure:
This is not an important step, If you can make an enclosure in your own style, it will be great and looks to be cool.We build a simple enclosure, to place the circuits is in secure before mounting it into the glass. We got a general purpose plastic enclosure from a nearby local store. Made somecuttings and holes for the ports on the Raspberry Pi. We painted them using some spray paints(black and white paints).And its look gorgeous!

b)Mounting To The Glass:

We have a Glass that laying around in our lab, Fortunately, it is the best glass that we can find to make this project perfect After placing everything on the glass, its look cool. We used a power bank, that we got recently from intel to power the raspberry Pi and its circuits.

III. CIRCUIT DIAGRAM & BLOCK DIAGRAM

The camera module is connected to the raspberry pi 3 camera port. The push button consists of 3 pins, one pin is connected to VSS, one pin is connected ground and another is connected to GPIO pin. The power supply is connected to raspberry pi 3 module. The speakers are also connected to raspberry pi 3.

When the person press the button on the Sight, The camera module will take a picture and send the image to android things , in android things the image will be access and analyze the image using tensor flow and detect what is that picture is about, then using a speaker or headphone, the Sight will voice assist the person about that picture.

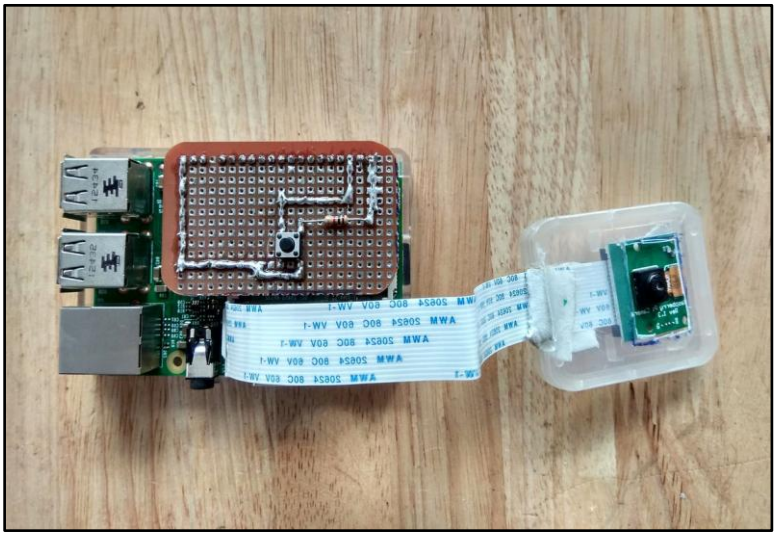

Figure.4 Circuit diagram

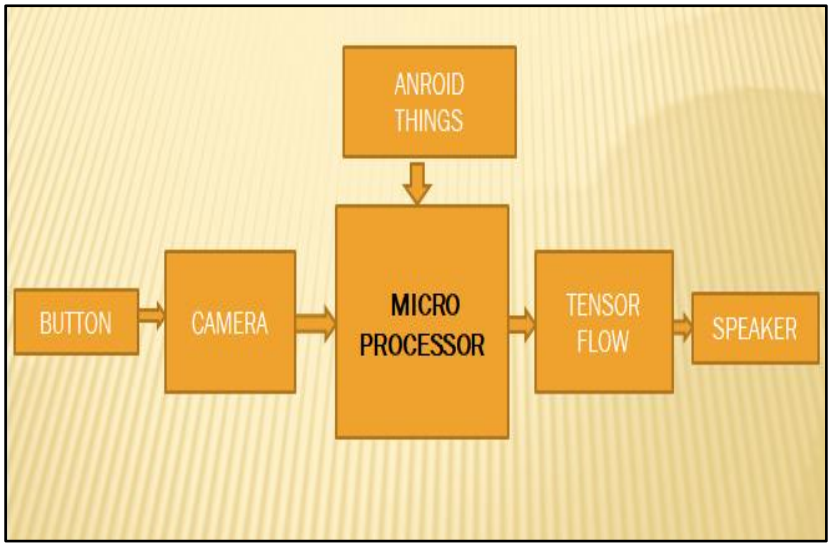

Figure.5 Block Diagram

When the person press the button on the Sight, The camera module will take a picture and send the image to android things , in android things the image will be access and analyze the image using tensor flow and detect what is that picture is about, then using a speaker or headphone, the Sight will voice assist the person about that picture.

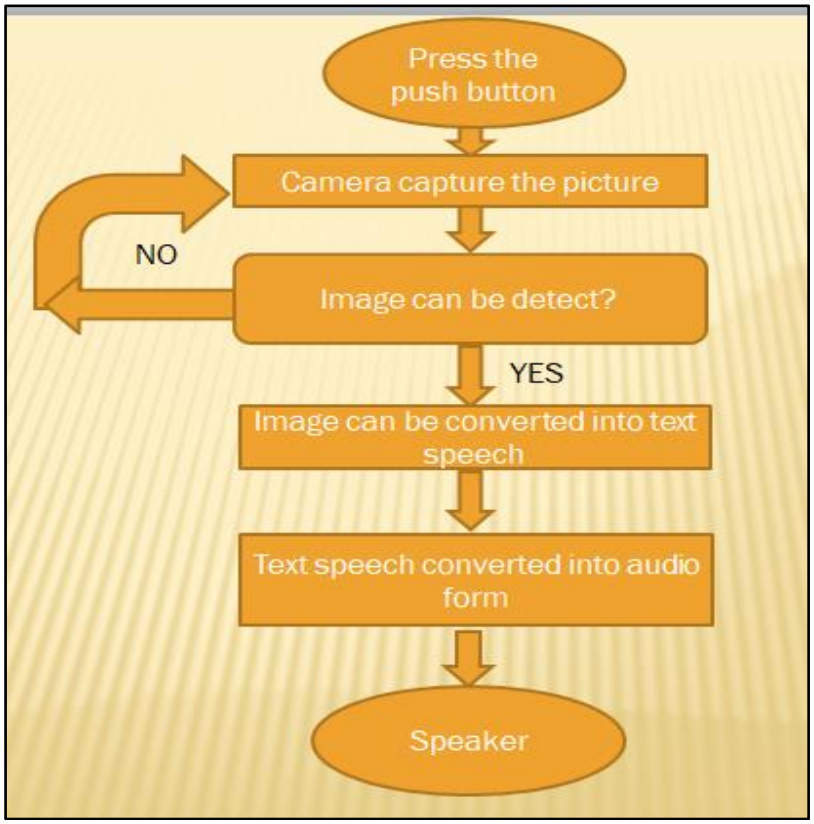

Figure.6 Flow chart

IV. CONCLUSION

This smart glass isa pair of glasses for the blind. By using smart glasses, a person can able to know what is going on in front of him/her. Smart glasses is developed with raspberry pi 3 and google android things. When the person press the button on the smart glasses, the camera will take a picture and analyze the image using tenser flow and detect what is that picture is about thenusing speaker or headphone, the google android with voice assist the person about the picture.It mainly used for blind people and also for military persons.

REFERENCES

[1]. Dr. P. G. Kuppusamy S. Kannan , S. Bharath Kumar, G. Arunraj, " Review of Light Fidelity Technology for Wireless Communication" International Journal of Applied Engineering Research, volume10,Research India publications.

[2]. Dr. P. G. Kuppusamy ,"An Artificial Intelligence Formulation and the Investigation of Glaucoma in Color Fundus Images by Using BAT Algorithm" Journal of Computational and Theoretical Nanoscience Vol. 14, 1–5, 2017.

[3]. P. G. Kuppusamy & R. Rani Hemamalini, "A VLSI Based Framework for Iterative and Adaptive Based Image Filter for Impulse Noise Removal" International Review on Computers and Software (I.RE.CO.S.), volume1,Pages235-242, Praise Worthy Prize.

[4]. P. G. Kuppusamy & R. Rani Hemamalini, "A Novel Iterative and Adaptive Noise Detection Method for Removal of Impulse Noise" International Journal of Engineering Research and Applications (IJERA),volume 3,issue 2.

[5]. V Prabhu, PG Kuppusamy, A Karthikeyan, M Sucharitha, "A novel approach for non-invasive measurement of mean arterial pressure using pulse transit time" Multimedia Tools and Applications 2018 Springer US.

[6]. P.Ratna Kamala , R.V.S.Satyanarayana, "Performance Analysis of Forward Error Correction Codes used for Transmission of data considering Failures in Coder/Decoder Circuits", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 6, June 2013.

[7]. P.Ratna Kamala , R.V.S.Satyanarayana, " Optimal Inner and Outer Error Correction Codes for Use in Space Communications: Performance Evaluation and Implementation". IUP Journal of Telecommunications; Hyderabad Vol. 8, Iss. 3, (Aug 2016): 13-26.

[8]. S. Dhananjeyan, Dr. K. MohanaSundaram , A. Kalaiyarasi and Dr. P. G. Kuppusamy, "Design and Development of Blind Navigation System using GSM and RFID Technology" Indian Journal of Science and Technology, Vol 9(2), DOI: 10.17485/ijst/2016/v9i2/85809, January 2016.
[9]. B.G.Gopal , P.G.Kuppusamy, "A Comparative Study on 4G and 5G Technology for Wireless Applications" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) .Volume 10, Issue 6, Ver. III (Nov - Dec .2015), PP 67-72.
[10]. V.Suresh KumarP.G.Kuppusamy and A.Karthikeyan, "IOT - Interoperability And Dynamic Workflow Composition – Challenges And Perspectives" Journal of electrical engineering, volume 9,Issue 1,Page nos.118-122.

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

Dr. P. G. Kuppusamy. "Smart Glasses for Blind People Based on ARM Embedded System Module." IOSR Journal of Engineering (IOSRJEN), vol. 09, no. 07, 2019, pp. 32-38.