Project Report

On

"Digital Escort – Your Personal Tourist Guide"

For

Renesas GR-Lychee Design Contest. 2018

Team ID: **Electronics Geek**
Company Name: **Marwadi Education Foundation, Rajkot**

Team Members
Bhavik Bhansali & Ravi Butani

YouTube Demo Video:

https://www.youtube.com/watch?v=bUA1KeDilRQ

Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ

Abstract:

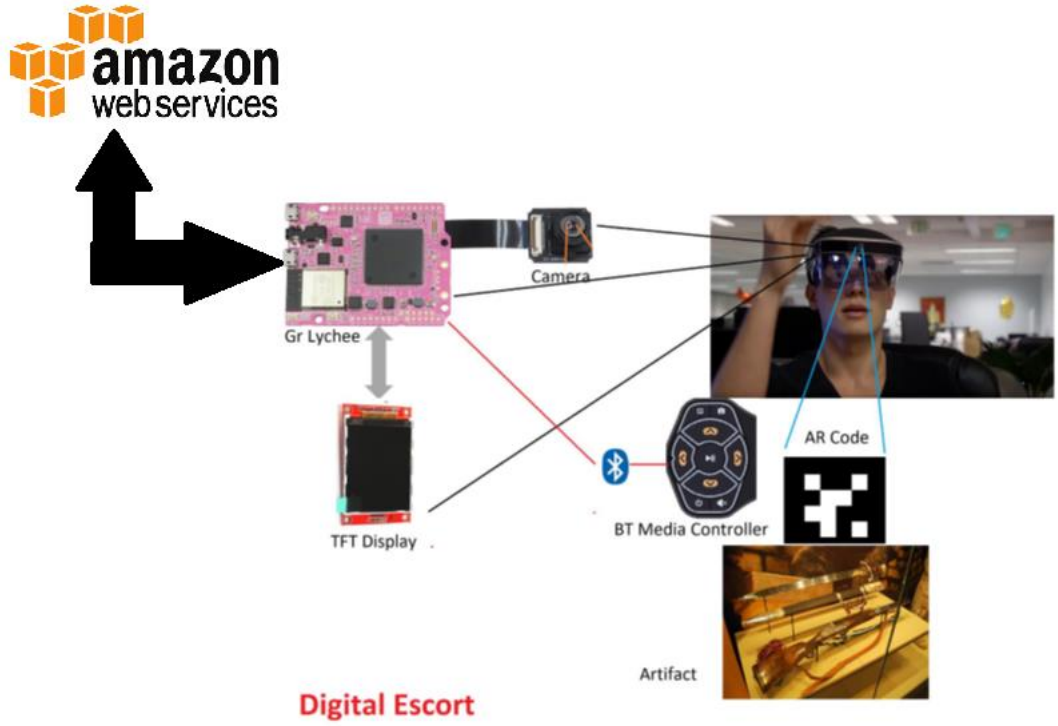
Project is mainly focus on the tourist for which India is in top 5 best tourist destination in the world and will generate 200Cr+ revenue from different tourist across India. Our proposed solution is to make AR glass to play the historical informative video/Audio of artefacts automatically when tourist see any artefacts. i.e. if you go to the museum and you see the sword of Maharan Pratap and you will get the historical informative video/audio on the glass you wear.

So, the idea is to place a set of AR (Augmented Reality) Marker codes on each artefact, when user of AR Glass sees the Artefacts The code going to be captured by camera and corresponding Audio/Video get played automatically.

In this project GR Lychee board is used as main processing unit to capture image using camera, find AR Code rectangle using open CV Shape Detection and send image to Amazon AWS EC2 instance by MQTT where Python code is running to do real time image processing for detecting the AR code. The ID of the AR code then sent to GR Lychee and show video/audio mapped with this code going to play using GR Lychee board only.

System Block Diagram:

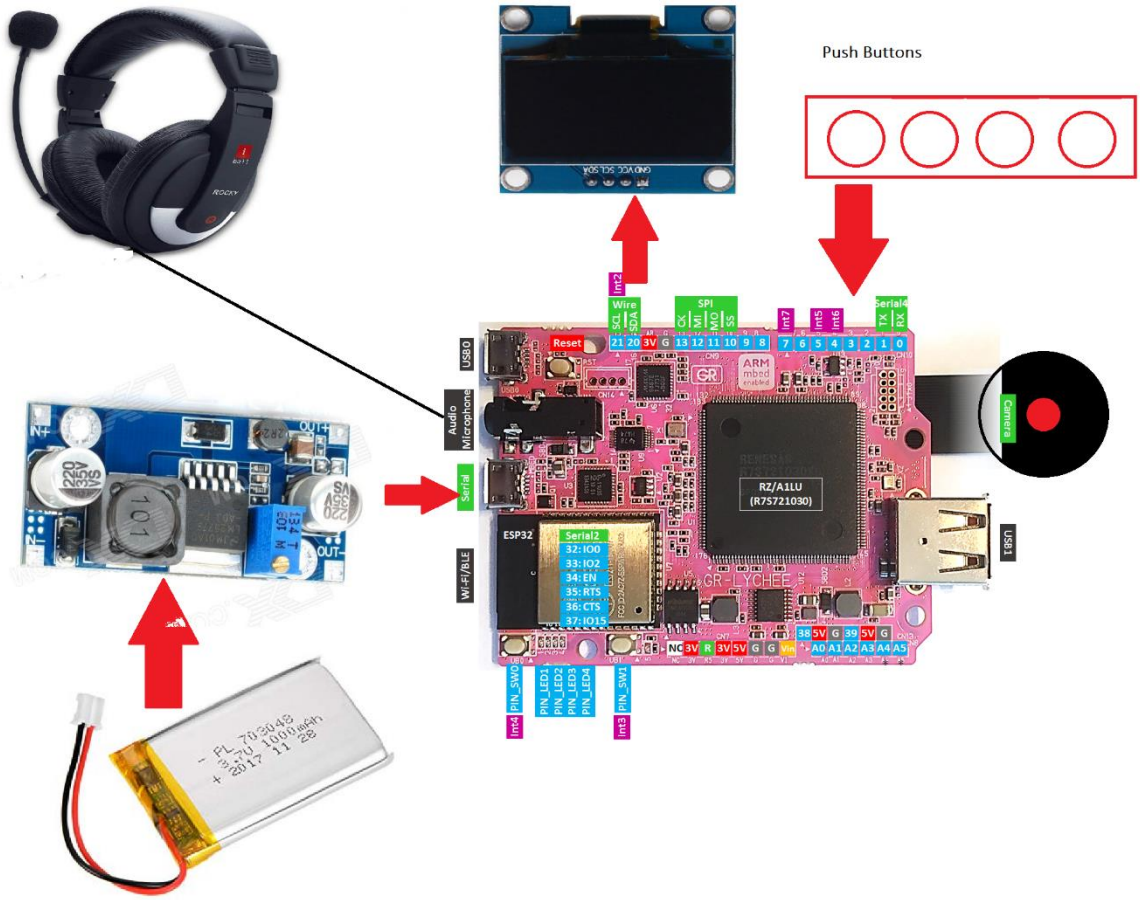
Here is a complete block diagram of designed project,



Team Members:

1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University

Hardware Connections:



Hardware Components:

1. **GR Lychee Board:** This is main processing unit for capture image, play audio, send image to MQTT Broker, Receive AR Code from MQTT Broker, Control 64x128 OLED Display and Camera.

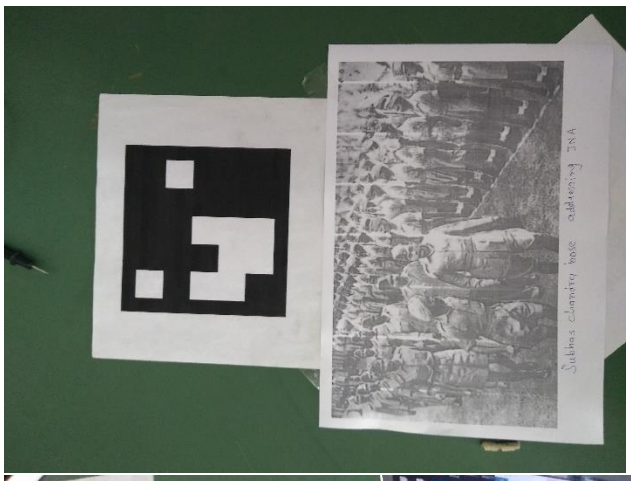
2. **3.7V 1000mAh LiPo Battery:** This battery provides power supply to GR Lychee and ther hardware vis 3,7V to 5V Boost Converter. Battery is capable to operate module for 7 Hour continue.

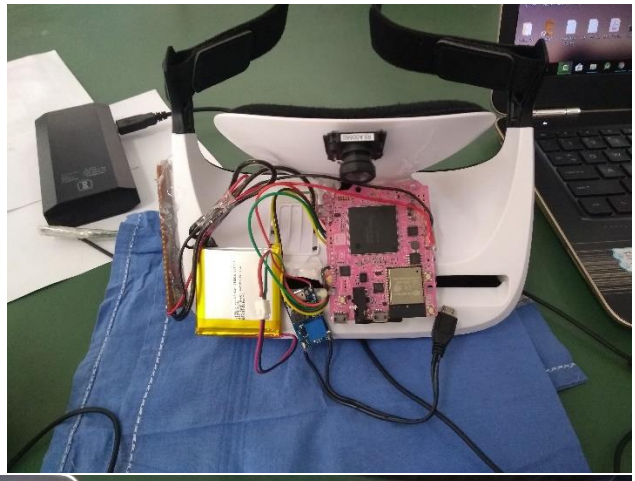
3. **64x128 OLED Display:** Flipped content on this display projected on AR Glass where it creates normal image for various information. Its connected to GR Lychee on I2C bus.

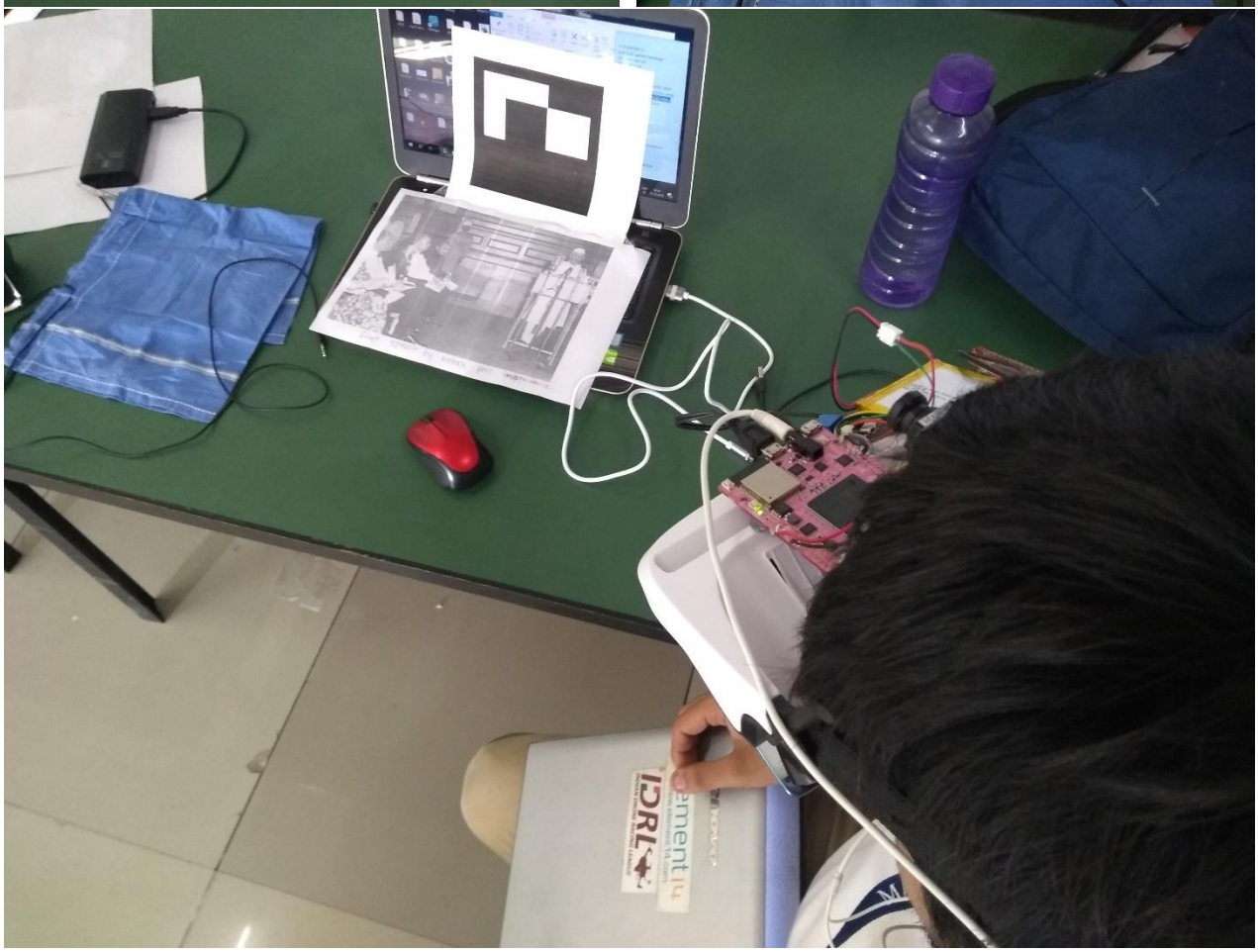
4. **GR Lychee Camera Module:** This 2Amp DC-DC boost converter converts 3.7V of battery in to 5V to provide power supply to the system.
5. **User Buttons:** These 4 push buttons with 10K pullup resistors contacted on D2,D3,D4 and D5 pins of GR Lychee to control Audio plyer i.e. Skip Audio, Pause/Resume Audio, Volume+ and Volume- operation.
6. **Audio Headphone:** Normal Audio Headphone is used to play Audio.

Hardware Setup: Here are hardware setup photos for Digital Escort Project,

Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ



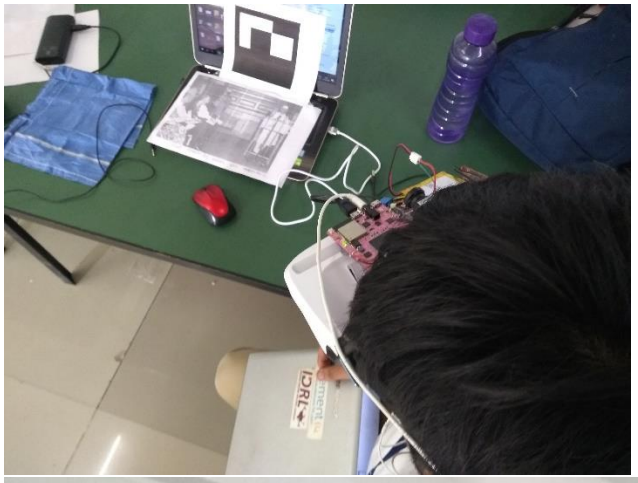




Team Members:

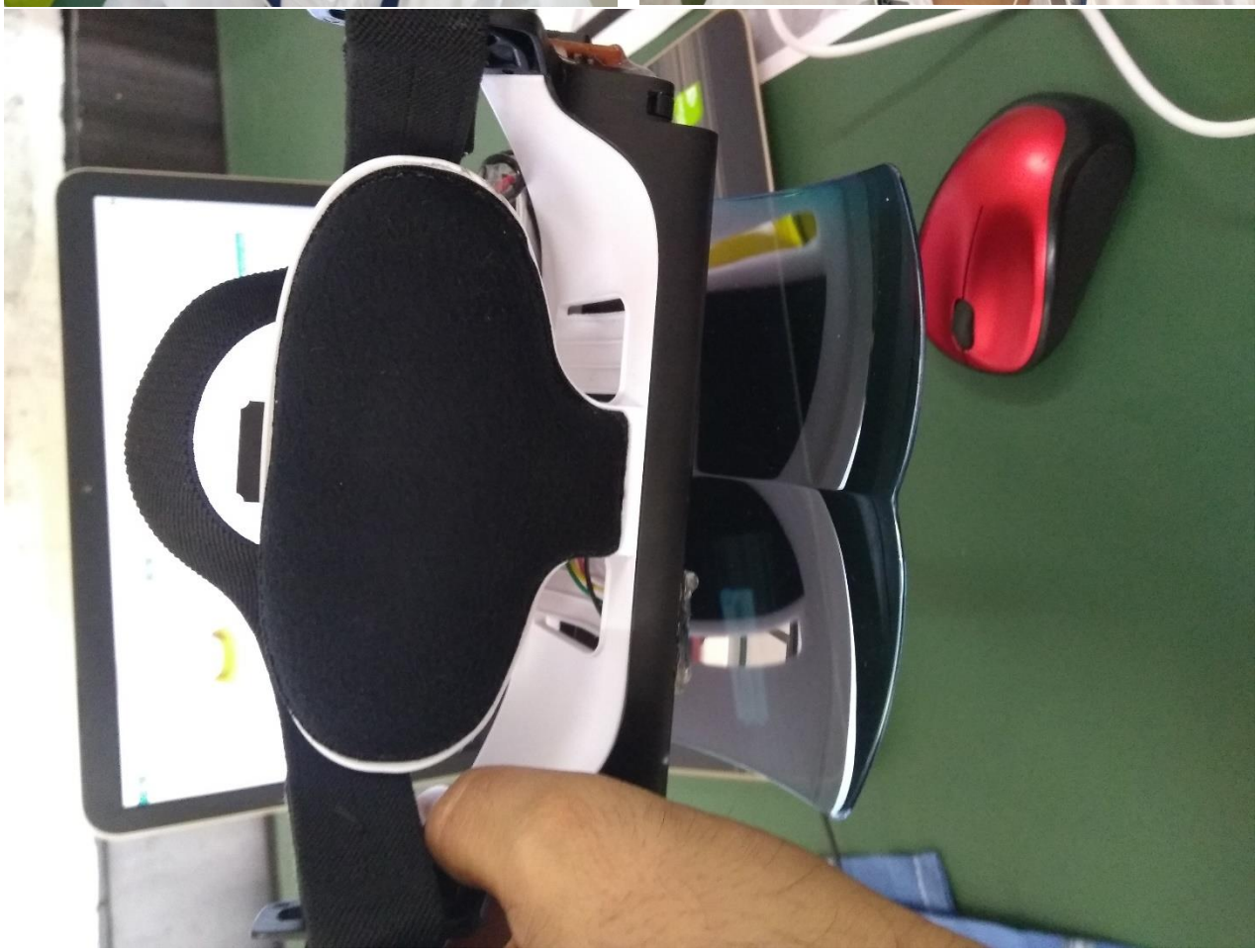
1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University

Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ



1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University

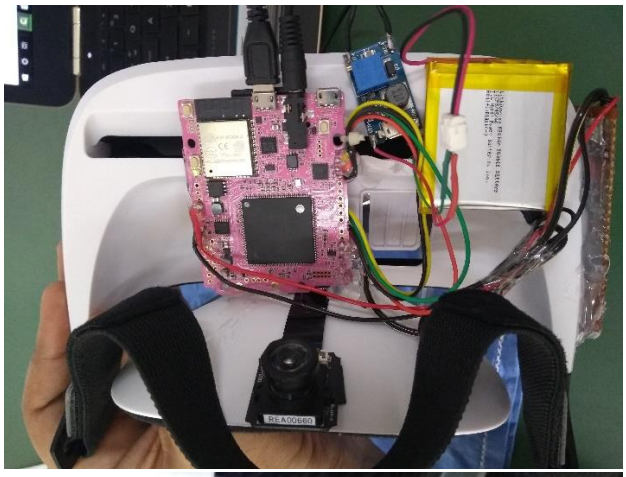
Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ



Team Members:

1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University

Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ

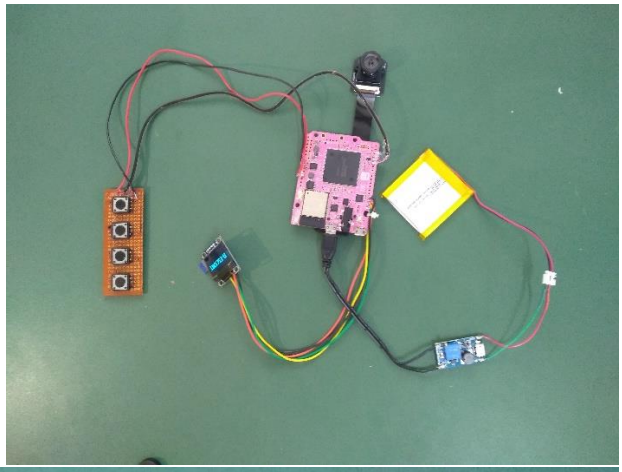


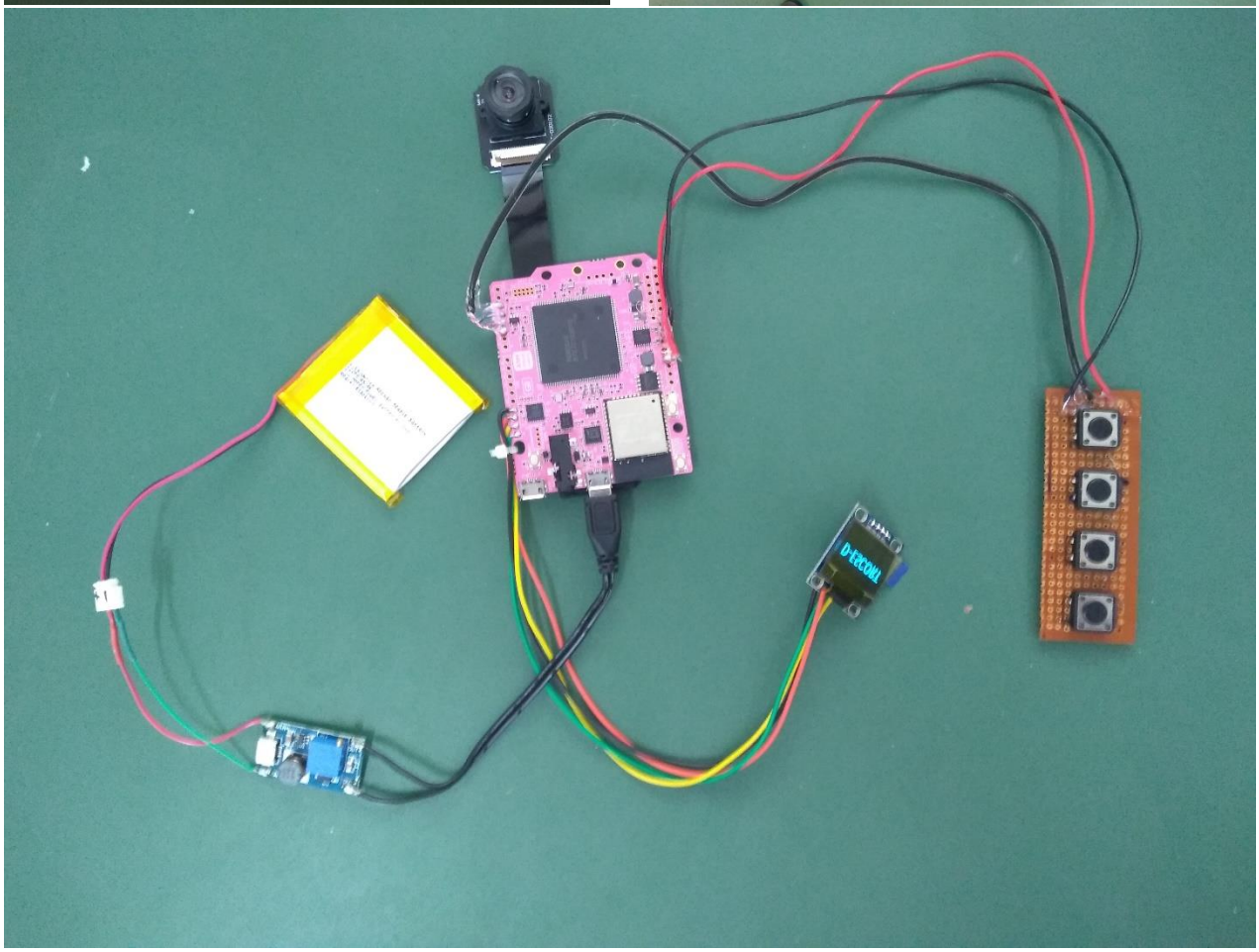
Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ

Team Members:

1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University

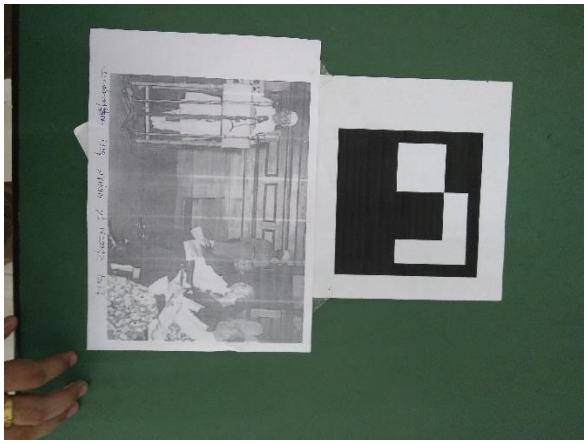
Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ

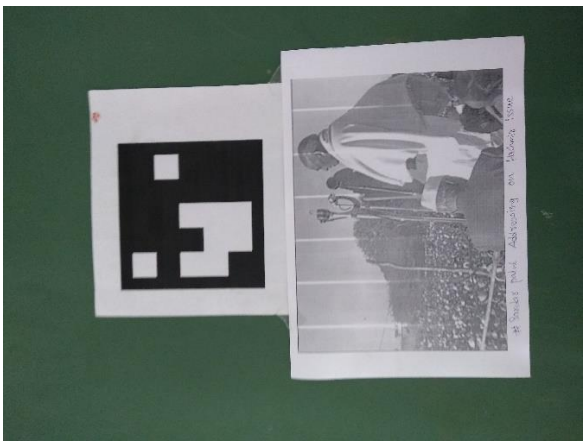




Team Members:

1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University





Working:

Software Codes:

Basically, there are two software developed for this project, first is firmware of GR Lychee and second is Python code running on Amazon AWS EC2 instance.

1. GR Lychee Code:

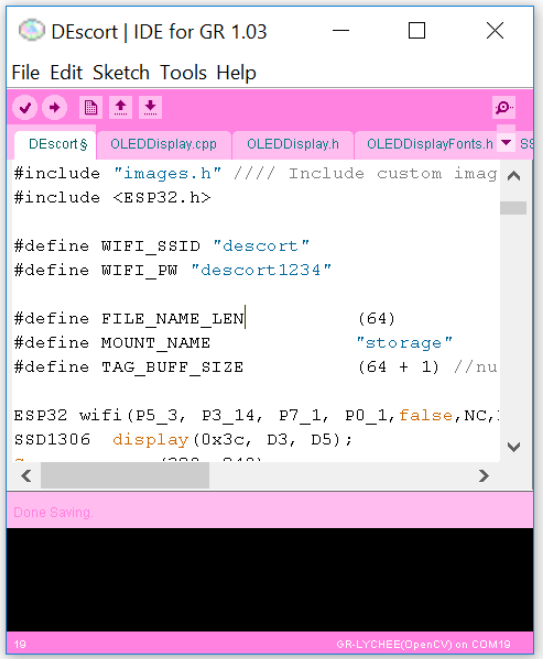
We have used GR-IDE for develop GR Lychee Firmware, we have used OLED Library for displaying content on 64x128 OLED, ESP32 Library with our own wrapper code for MQTT Publish Subscribe and Easyplayback library for plying wav audio file.

MQTT Broker: our own Broker Hosted on Amazon AWS EC2 instance

Publish Topic: arcodepub

Subscribe Topic: arcodesub

Team ID: Electronics Geek
Title: Digital Escort – Your Personal Tourist Guide
Company Name: Marwadi Education Foundation, Rajkot
YouTube Demo Video: https://www.youtube.com/watch?v=bUA1KeDilRQ




```
// Digital Escort GR Lychee Code
#include "Arduino.h"
#include "mbed.h"
#include "SdUsbConnect.h"
#include "EasyPlayback.h"
#include "EasyDec_WavCnv2ch.h"
#include <Camera.h>
#include <SD.h>
#include <SPI.h>
#include <RTC.h>
#include <Wire.h>  // Only needed for Arduino 1.6.5 and earlier
#include "SSD1306.h" // alias for `#include "SSD1306Wire.h"`
#include "images.h" // Include custom images
#include <ESP32.h>

#define WIFI_SSID "descort" // ssid
#define WIFI_PW "descort1234" // password

#define FILE_NAME_LEN        (64)
#define MOUNT_NAME           "storage"
#define TAG_BUFF_SIZE        (64 + 1) //null-terminated

ESP32 wifi(P5_3, P3_14, P7_1, P0_1,false,NC,NC,1000000);
SSD1306  display(0x3c, D3, D5);
Camera camera(320, 240);
static float vol =1.0;
static int aud_flag = 0;
static int marker = 0;
static InterruptIn skip_btn(D2);
static InterruptIn pause_btn(D4);
static InterruptIn vol_up_btn(D3);
```


Team Members:

1. Bhavik Bhansali - Research Scholar, Marwadi University
2. Ravi Butani - 1st Sem PhD Student, Marwadi University

```
static EasyPlayback AudioPlayer;

byte connectpkt[] = {0x10,0x12,0x00,0x04,'M','Q','T','T',0x04,0x02,0x00,0xff,0x00,0x06,'r','2','u','z','m','e'};
byte pubpkt[]={0x30, 0x14, 0x00, 0x09 ,'r','a','v','i','m','q','t','t','h','e','l','l','o','i','r','a','v','v'};
byte subpkt[]={0x82,0x0E,0x00,0x01,0x00,0x09,'a','r','c','o','d','e','p','u','b',0x00};
char pngpkt[]={0xC0,0x00};
static byte pubimg[35000];
int resp;
File myFile;
int timep = 1;

static void vol_up_btn_fall(void) {
   if(vol <0.9) vol = vol+0.1;
   AudioPlayer.outputVolume(vol);
   Serial.println("up");
   //delay(200);
}

static void vol_dn_btn_fall(void) {
   if(vol>0.1) vol = vol-0.1;
   AudioPlayer.outputVolume(vol);
   Serial.println("dn");
   delay(200);
}

static void pause_btn_fall(void) {
   AudioPlayer.pause();
   delay(200);
   Serial.println("ps");
}

static void skip_btn_fall(void) {
   AudioPlayer.skip();
   Serial.println("skp");
   delay(200);
   display.clear();
   display.display();
   aud_flag == 0;
}

void setup(void) {
 Serial.begin(115200);
 display.init();
 display.flipScreenVertically();
 display.setFont(ArialMT_Plain_10);
 display.clear();
 display.drawXbm(0, 16, scr_w ,scr_h, scr );
 display.display();
 delay(3000);

 SdUsbConnect storage(MOUNT_NAME);
```

```
// decoder setting
AudioPlayer.add_decoder<EasyDec_WavCnv2ch>(".wav");
AudioPlayer.add_decoder<EasyDec_WavCnv2ch>(".WAV");
// volume control
AudioPlayer.outputVolume(0.7);  // Volume control (min:0.0 max:1.0)

// button setting
skip_btn.fall(&skip_btn_fall);
pause_btn.fall(&pause_btn_fall);
vol_up_btn.fall(&vol_up_btn_fall);
vol_dn_btn.fall(&vol_dn_btn_fall);

// wait for the storage device to be connected
printf("Finding a storage...\r\n");
storage.wait_connect();
printf("done\r\n");

camera.begin();
delay(100);
SD.begin();
Serial.println("Starts.");
Serial.print("Connecting Wi-Fi..");
wifi.connect(WIFI_SSID, WIFI_PW);
Serial.println("done");
Serial.println(wifi.getIPAddress());
delay(1000);
Serial.print("Broker :");
resp = wifi.open("TCP",0,"iot.eclipse.org", 1883);
resp = wifi.open("TCP",0,"35.170.211.125", 1883);
resp = wifi.open("TCP",0,"192.168.43.136", 1883);
Serial.println(resp);
delay(100);
Serial.print("Broker : ");
resp = wifi.send(0,connectpkt, 20);
Serial.println(resp);
delay(100);
Serial.print("Sub : ");
resp = wifi.send(0,subpkt, 16);
Serial.println(resp);
delay(100);
AudioPlayer.play("/storage/5.wav");
delay(3000);
while(1)
{
  unsigned int len;
  unsigned int point = 0;
  int siz = 0;
  delay(10);
  pubimg[0] = 0x30;
  pubimg[1] = 0x84;
  pubimg[2] = 0x2A;
```

```
  pubimg[3] = 0x00;
  pubimg[4] = 0x08;
  pubimg[5] = 'a';
  pubimg[6] = 'r';
  pubimg[7] = 'i';
  pubimg[8] = 'm';
  pubimg[9] = 'g';
  pubimg[10] = 's';
  pubimg[11] = 'u';
  pubimg[12] = 'b';
  point = 13;
  // read from the camera until there's nothing else in it:
 /* size_t size = camera.createJpeg();
  delay(100);
  uint8_t* adr = camera.getJpegAdr();
   for (size_t i = 0; i < size; i++) {
     pubimg[point]=*adr;
     adr++;
     point++;
   }
  delay(100);
  len = size+10;
  pubimg[1]=(len%128)|0x80;
  pubimg[2]=(len/128);
  // read from the file until there's nothing else in it:
  Serial.print("pubimg : ");
  //resp = wifi.send(0,pubimg,5383);
  resp = wifi.send(0,pubimg,len+3);
  Serial.println(resp);*/
  if(resp == 0)
  {
   wifi.close(0,2);
   delay(100);
   Serial.print("Reconnect : ");
   resp = wifi.open("TCP",0,"192.168.43.136", 1883);
   Serial.println(resp);
   delay(100);
   Serial.print("Broker : ");
   resp = wifi.send(0,connectpkt, 20);
   Serial.println(resp);
   delay(100);
   Serial.print("Sub : ");
   resp = wifi.send(0,subpkt, 16);
   Serial.println(resp);
  }
  while(wifi.readable())
  {
   char rxbuffer[]={"******************************"};
   delay(10);
   resp = wifi.recv(0,rxbuffer, sizeof rxbuffer);
   if(rxbuffer[13] != '*')
```

```
    {
      display.clear();
      display.drawXbm(0, 16, det_w ,det_h, det );
      display.display();
      Serial.print("msg = ");
      Serial.write(rxbuffer[13]);
      Serial.write(rxbuffer[14]);
      Serial.write(rxbuffer[15]);
      Serial.write(rxbuffer[16]);
      Serial.println();
      AudioPlayer.play("/storage/2.wav");
      aud_flag=1;
      if(rxbuffer[13] == '1')AudioPlayer.play("/storage/1.wav");
      else if(rxbuffer[13] == '2')AudioPlayer.play("/storage/2.wav");
      else if(rxbuffer[13] == '3')AudioPlayer.play("/storage/3.wav");
      else if(rxbuffer[13] == '4')AudioPlayer.play("/storage/4.wav");
      while(aud_flag == 1);
     }
    }
    if(timep%10 == 0)
    {
     Serial.println("ping");
     resp = wifi.send(0,pngpkt, 2);
    }
    timep++;
    delay(500);
}
}

void loop() {
// Nothing Here , Execution never reached here due to infinite loop in setup for considering scope of AudioPlayer
//class
}
```

2. Python Code:

We have developed Python Code which Running on EC2 Instance of Amazon AWS to receive images from GR Lychee over MQTT protocol process the image to detect AR Marker and send code of AR Marker back to GR Lychee over MQTT Protocol. Here is a code for reference,

```
#!/usr/bin/env python
#--------------------------------------------------------------------------------
# GR Digital Escort
# Python Script running on Amazon AWS EC2 to receive images over MQTT in realtime
# Process Image to detect AR Code and send back AR Code ID to GR Lychee
#--------------------------------------------------------------------------------
import paho.mqtt.client as mqttClient
import time
import cv2
```

```
import numpy as np
from ar_markers.hamming.detect import detect_markers
fn = 1
pro_flag = 0
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to broker")
        global Connected                #Use global variable
        Connected = True                #Signal connection
    else:
        print("Connection failed")

def on_message(client, userdata, message):
    global fn
    global pro_flag
    print "Message received: "
    pro_flag = 1
    f = open("test.jpg", "wb")
    f.write(message.payload)
    f.close()
    frame = cv2.imread("test.jpg",cv2.IMREAD_COLOR )
    markers = detect_markers(frame)
    for marker in markers:
        marker.highlite_marker(frame)
        print 'marker.center'
        print marker.center
        print 'marker.contours'
        print cv2.contourArea(marker.contours)
        print 'marker.id'
        print marker.id
        client.publish("arcodepub",str(marker.id))

Connected = False   #global variable for the state of the connection
broker_address= "35.170.211.125"  #Broker address of Amazon AWS
port = 1883                       #Broker port

client = mqttClient.Client("Python")               #create new instance
client.on_connect= on_connect                      #attach function to callback
client.on_message= on_message                      #attach function to callback
client.connect(broker_address, port=port)          #connect to broker
client.loop_start()        #start the loop

while Connected != True:    #Wait for connection
    time.sleep(0.1)
client.subscribe("imagesub")

client.subscribe("imagesub")
try:
    while True:
        time.sleep(0.1)
except KeyboardInterrupt:
    print "exiting"
    client.disconnect()
    client.loop_stop()
```

```
# When everything done, release the capture
capture.release()
cv2.destroyAllWindows()
```

Conclusion:

With this project, we have gained lots of hands-on knowledge about GR Lychee Board programming and OpenCV programming. We have completed content display part od Digital Escort project and working on Indoor navigation part using similar AR Code principle.