Team Name : SOVEREIGNS

Team Members : \* G.Anjali

\* Durga.V

\* B.Kalyan

ABSTRACT :

Billboard [advertisements](https://www.boredpanda.com/powerful-social-advertisements/) are designed to catch a person’s attention and create a memorable impression very quickly, leaving the reader thinking about the ad after they have driven past it. The advertising ideas behind it have to readable in a very short time because they are usually read while being passed at high speeds. Thus there are generally only a few words, in large print, and a humorous or arresting image in brilliant color.

Besides targeting a specific audience according to location, billboard [ads](https://www.boredpanda.com/365-days-of-print-ads-its-been-128-days-ive-been-making-print-ads-every-single-day-still-237-days-to-go/) are also on 24 hours a day. Compare the 24 hours per day exposure of a billboard versus a TV [commercial](https://www.boredpanda.com/15-extremely-creative-durex-condom-ads/) or an advertisement print in a newspaper. These ads have a useful life of 30 seconds at most. In addition to this great advantage is the fact that billboard advertising is very affordable.

Although some of the top billboards are alongside highways, there are many billboards on buildings in a city space where viewers can spend much more time reading the marketing campaign message. So it’s always important to take the context (where the ad appears) into consideration.

Companies can attract the customers by doing advertisements. These smart bill boards will help them in attracting their customers and make their task easier. In this we can upload the required data on the bill board simply by giving inputs through user interface. And we can check the lamps working status which is connected to bill board through the UI.

Through device we can select the mode of the display and according to the selected mode we can change the data on the display. the data can be entered through user interface which is created in Node Red. We can get the status of the lamps which are connected to the bill boards in the user interface. If any lamp fails we can send notifications to authorities.

Introduction: SMART BILLBOARDS

In the past, billboards were created using sections of paper pasted together to form one image. Nowadays, digital printing on vinyl material allows the entire image to be printed on one sheet and can be erected very quickly and easily. Digital billboards allow creative executions that can be personalized, or swapped out with other creative ads. Sometimes, clients can choose to timeshare their ads with other companies, with the creative swapping out every few minutes.

**Smart billboards for smarter audiences**

 One of the key features of smart billboards is the ability to deliver relevant and timely content that is highly effective and enable engagement with the end user. Advertisers have realized the importance of contextual relevance when it comes to attracting audiences who are now digital savvy or consumers who are always on the move with the means of out-of-home advertising.

In today’s day and time, individuals have become prone to using multiple digital platforms to consume their daily information. Digital screens are set with the ability to provide richer content that is contextually relevant and in sync with consumers’ daily interests while  ensuring interactivity at the same time.The rise of contextual messaging that is powered by data on location, weather updates, traffic, breaking news, etc., has given brands an opportunity to tailor their content to stay relevant. It is essential for us to realize that every brand needs to evolve and innovate.

Smart billboards provide another unique opportunity for brands to create personalized content on the basis of a person’s mood, reaction, geo fencing, etc. Imagine, sensors are used to trigger an ad of Kaun Banega Crorepati (the Indian version of “Who Wants to be a Millionaire”) on a nearby screen, prompting a message ‘You could be our next Crorepati’ and your image is placed next to the host. At the same time, the screen showcases a number or code you can scan that will lead you to the website.Once the content and visuals are in place, the location makes a lot of difference. Advertisers need to initiate a footfall analysis where they can understand what location makes a better impact through the same billboard.

HARDWARE COMPONENTS:

1.NODEMCU :



|  |  |
| --- | --- |
| NodeMCU DEVKIT 1.0 | |
| **Developer** | ESP8266 Opensource Community |
| **Type** | [Single-board microcontroller](https://en.wikipedia.org/wiki/Single-board_microcontroller) |
| **Introductory price** | $5 |
| [**Operating system**](https://en.wikipedia.org/wiki/Operating_system) | XTOS |
| [**CPU**](https://en.wikipedia.org/wiki/Central_processing_unit) | ESP8266[[1]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-1)(LX106[[2]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-8266sdk-2)) |
| **Memory** | 128kBytes |
| **Storage** | 4MBytes[[3]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-3) |
| **Power** | USB |

**NodeMCU** is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things" \o "Internet of Things)platform.[[4]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-nodemcu_firmware-4)[[5]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-5) It includes [firmware](https://en.wikipedia.org/wiki/Firmware" \o "Firmware)which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi" \o "Wi-Fi)[SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from Espressif Systems, and hardware which is based on the ESP-12 module.[[6]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-Espressif_Systems-6)[[7]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-7) The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)" \o "Lua (programming language)) scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson[[8]](https://en.wikipedia.org/wiki/NodeMCU" \l "cite_note-8) and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1).[[9]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-spiffs-9)

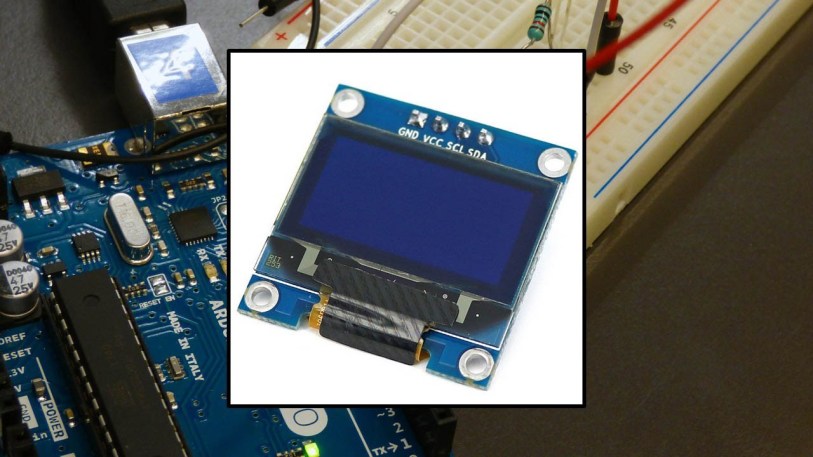
History:

NodeMCU was created shortly after the [ESP8266](https://en.wikipedia.org/wiki/ESP8266)came out. On December 30, 2013, [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1" \o "Espressif Systems (page does not exist))[[6]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-Espressif_Systems-6) began production of the ESP8266.[[10]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-10)The ESP8266 is a Wi-Fi SoC integrated with a [Tensilica](https://en.wikipedia.org/wiki/Tensilica" \o "Tensilica) Xtensa LX106 core,[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] widely used in IoT applications (see [related projects](https://en.wikipedia.org/wiki/NodeMCU#Related_projects)). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub.[[11]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-11) Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the [gerber](https://en.wikipedia.org/wiki/Gerber_format" \o "Gerber format) file of an ESP8266 board, named devkit v0.9.[[12]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-init_devkit-12) Later that month, Tuan PM ported [MQTT](https://en.wikipedia.org/wiki/MQTT) client library from [Contiki](https://en.wikipedia.org/wiki/Contiki" \o "Contiki) to the ESP8266 SoC platform,[[13]](https://en.wikipedia.org/wiki/NodeMCU" \l "cite_note-mqtt_client-13) and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib[[14]](https://en.wikipedia.org/wiki/NodeMCU" \l "cite_note-u8g-14) to NodeMCU project,[[15]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-u8glib-15) enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.



2.OLED(ORGANIC LIGHT EMITTING DIODE):

The organic light-emitting diode (OLED) display that we’ll use in this tutorial is the SSD1306 model: a monocolor, 0.96-inch display with 128×64 pixels as shown in the below figure.



The OLED display doesn’t require backlight, which results in a very nice contrast in dark environments. Additionally, its pixels consume energy only when they are on, so the OLED display consumes less power when compared with other displays.

The model we’re using here has only four pins and communicates with the Arduino using I2C communication protocol. There are models that come with an extra RESET pin. There are also other OLED displays that communicate using SPI communication.

**Pin wiring**

Because the OLED display uses I2C communication protocol, wiring is very simple. You just need to connect to the Arduino Uno I2C pins as shown in the table below.

|  |  |
| --- | --- |
| **Pin** | **Wiring to Arduino Uno** |
| Vin | 3.3V |
| GND | GN GNDVD |
| SCL | A5 |
| SDA |  |

The Adafruit library for the OLED display comes with several functions to write text. You can display 128×64 bitmap monocolor images on the OLED display.

It can also displays temperature and humidity readings on the OLED display. We’ll get temperature and humidity using the [DHT11 temperature and humidity sensor](https://makeradvisor.com/tools/dht11-temperature-humidity-sensor/).

3.BUTTONS:



When the pushbutton is **open** (unpressed) there is **no connection** between the two legs of the pushbutton, so the pin is connected to 3.3v (through the pull-up resistor) and we read a **HIGH**.

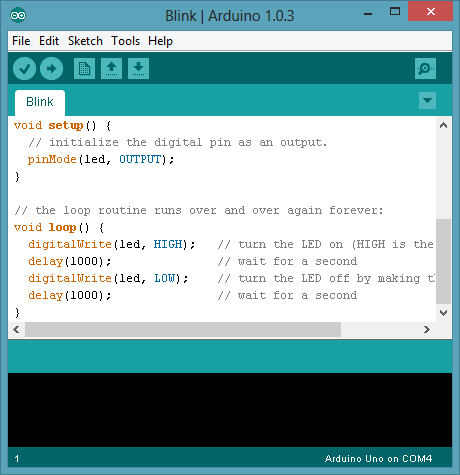
When the pushbutton is**closed**(pressed), there is **a connection** between its two legs, connecting the pin to ground, so that we read a **LOW**. (The pin is still connected to 5 volts, but the resistor in-between them means that the pin is "closer" to ground)

SOFTWARE COMPONENTS:

1.ARDUINO IDE:

1. Arduino first and foremost is an open-source computer hardware and software company. The Arduino Communityrefers to the project and user community that designs and utilizes microcontroller-based development boards. These development boards are known as Arduino Modules, which are open-source prototyping platforms. The simplified microcontroller board comes in a variety of development board packages.
2. The most common programming approach is to use the Arduino IDE, which utilizes the C programming language. This gives you access to an enormous Arduino Library that is constantly growing thanks to open-source community.
3. Arduino IDE:

The Arduino IDE once it’s been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us touploadcode. Connect your Arduino board to the PC via the USB cablE touploadcode. Connect your Arduino board to the PC via the USB cable.



2.IBM WATSON CLOUD PLATFORM:

Watson was created as a [question answering](https://en.wikipedia.org/wiki/Question_answering) (QA) computing system that IBM built to apply advanced [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval), [knowledge representation](https://en.wikipedia.org/wiki/Knowledge_representation), [automated reasoning](https://en.wikipedia.org/wiki/Automated_reasoning), and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technologies to the field of [open domain question answering](https://en.wikipedia.org/wiki/Open_domain_question_answering).[[2]](https://en.wikipedia.org/wiki/Watson_(computer)#cite_note-ibm-2)

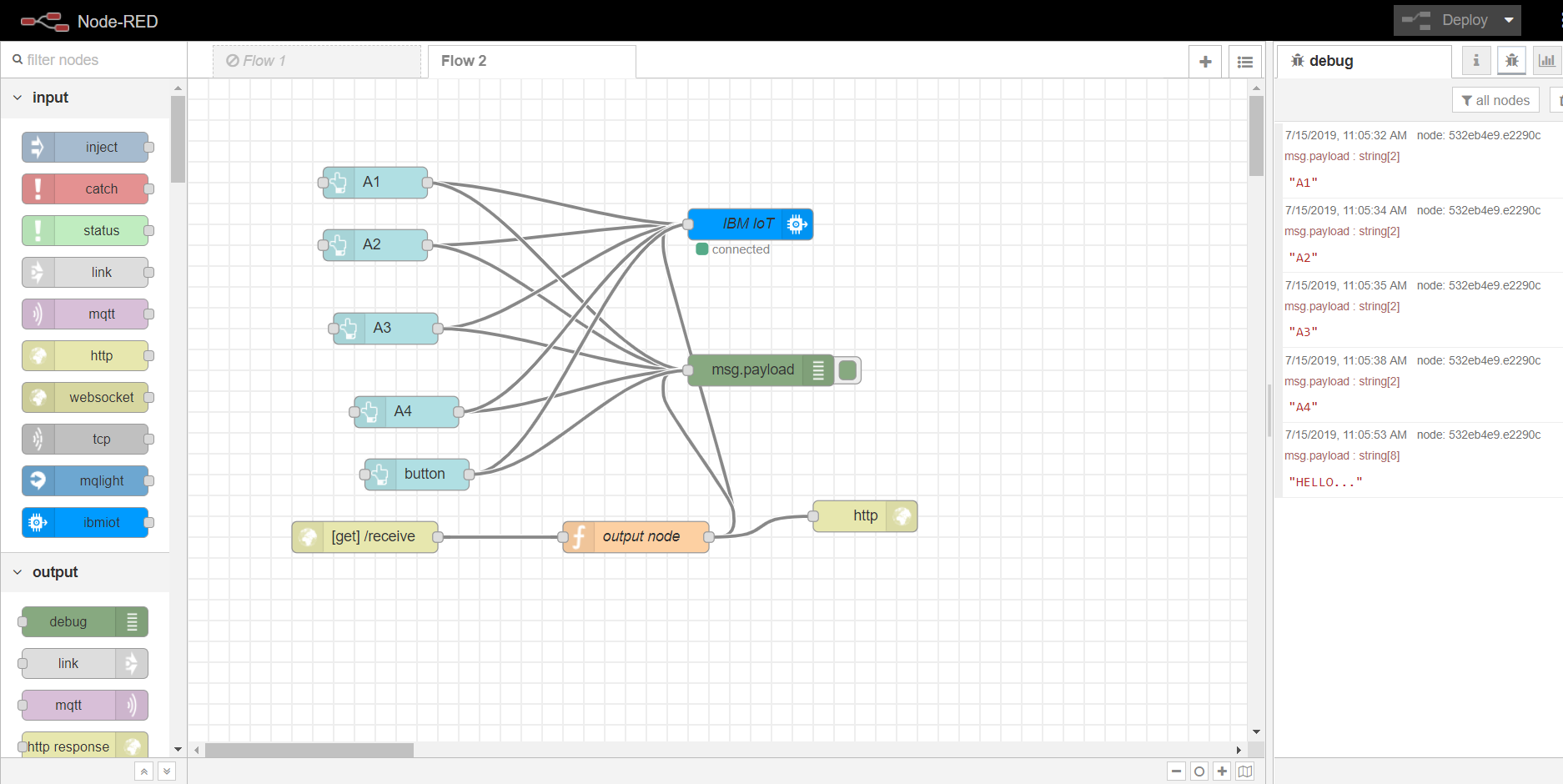
The key difference between QA technology and document search is that document search takes a keyword query and returns a list of documents, ranked in order of relevance to the query (often based on popularity and page ranking), while QA technology takes a question expressed in natural language, seeks to understand it in much greater detail, and returns a precise answer to the question.[[12]](https://en.wikipedia.org/wiki/Watson_(computer)#cite_note-12)

When created, IBM stated that, "more than 100 different techniques are used to analyze natural language, identify sources, find and generate hypotheses, find and score evidence, and merge and rank hypotheses."[[13]](https://en.wikipedia.org/wiki/Watson_(computer)#cite_note-13)

In recent years, the Watson capabilities have been extended and the way in which Watson works has been changed to take advantage of new deployment models (Watson on IBM Cloud) and evolved machine learning capabilities and optimised hardware available to developers and researchers. It is no longer purely a [question answering](https://en.wikipedia.org/wiki/Question_answering) (QA) computing system designed from Q&A pairs but can now 'see', 'hear', 'read', 'talk', 'taste', 'interpret', 'learn' and 'recommend'.



OUTPUTS:

1.NODERED:

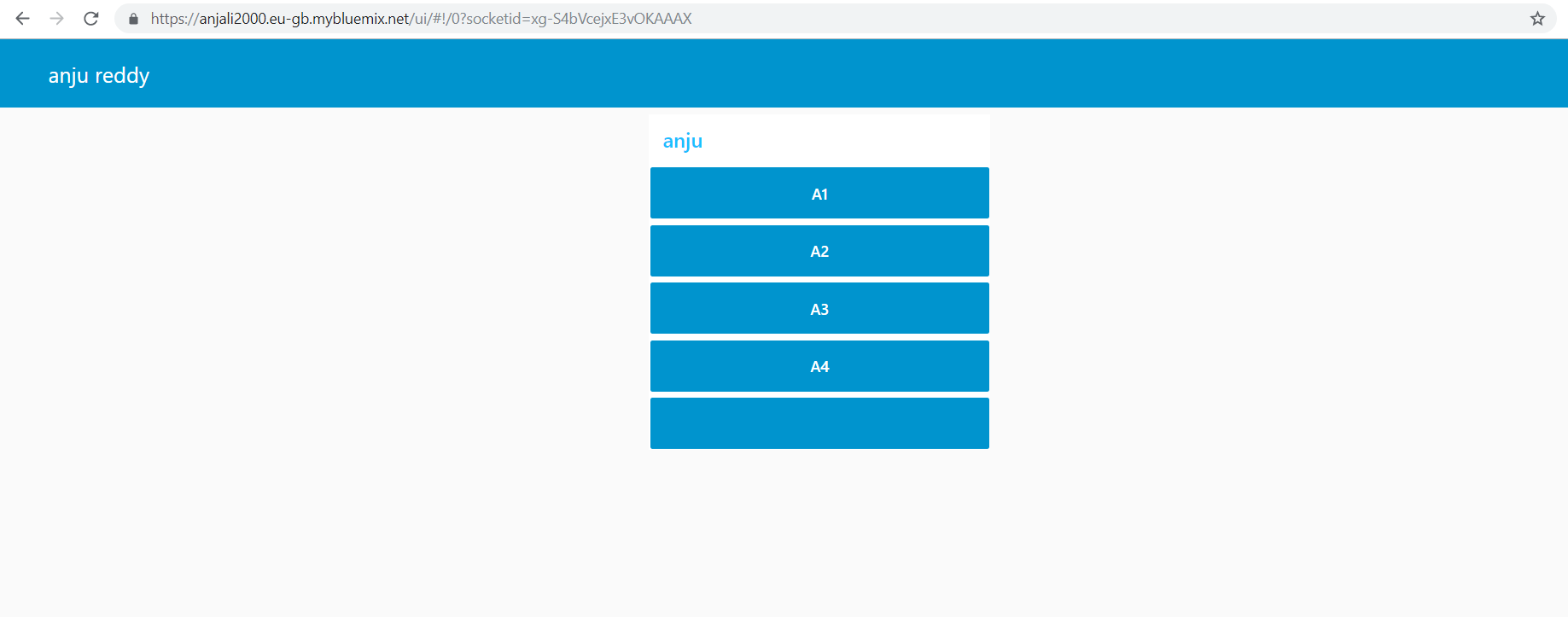
There are several ways to interact with an Arduino using Node-RED. They all assume the Arduino is connected to the host computer via a USB serial connection.

Node-RED is a tool for wiring together hardware devices, APIs and online services in new and interesting ways.

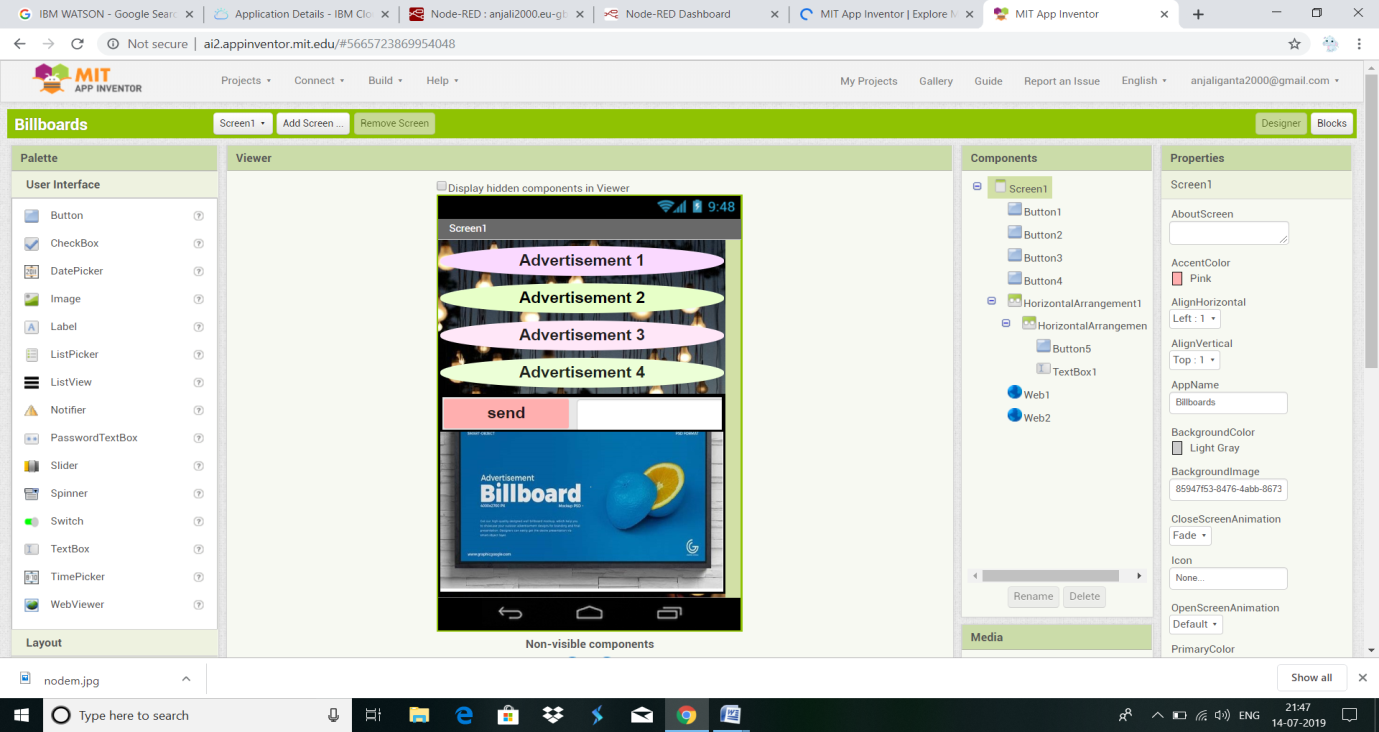
Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range nodes in the palette. Flows can be then deployed to the runtime in a single-click.

**2.USER INTERFACE(UI) IMAGE:**

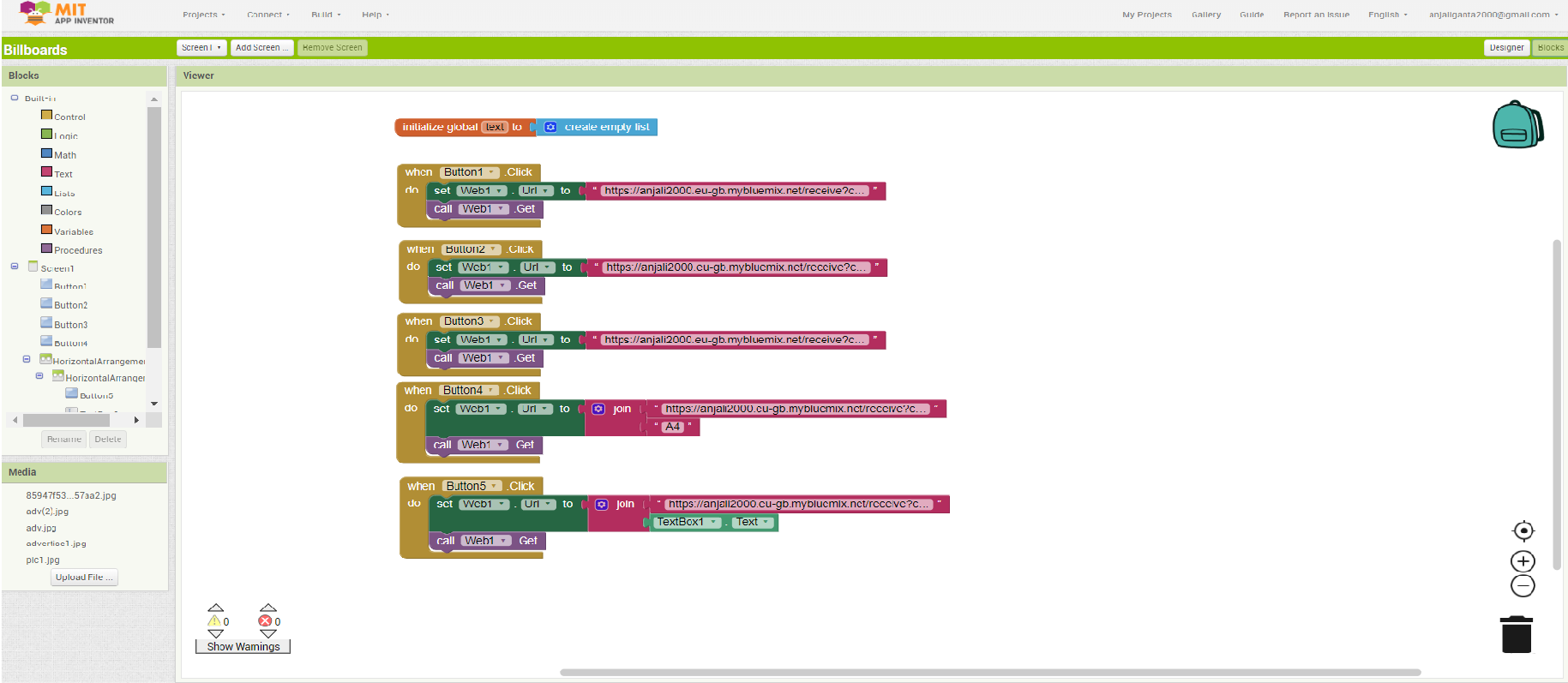
The dashboard or display nodes appear on the **User Interface** (**UI**) dashboard. An example **UI** dashboard is shown below: You can have multiple display pages called tabs and each tab has a name. On a Page the display nodes can be arranged in Groups.

****3.MIT APP:

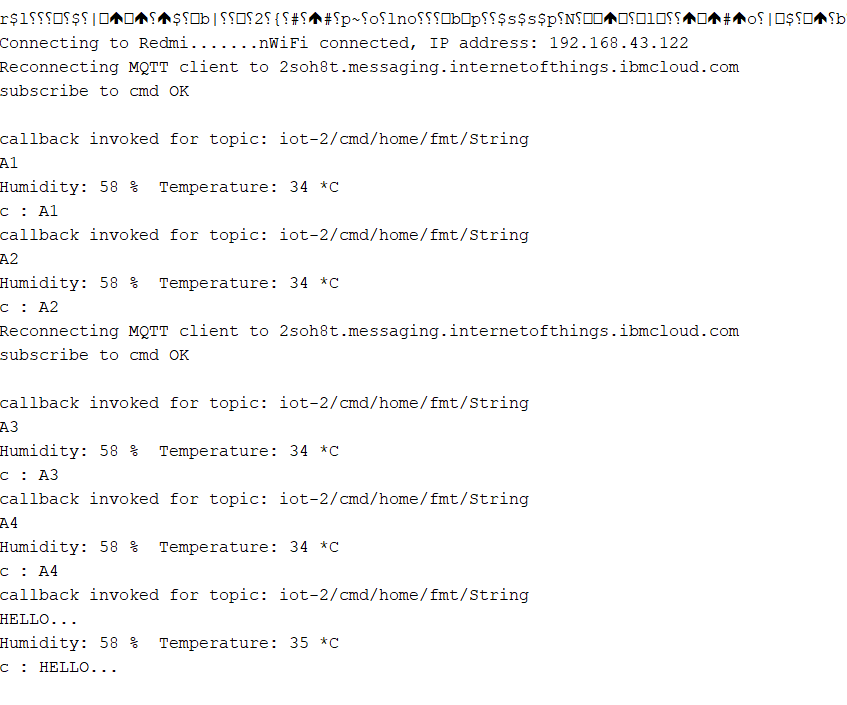
**MIT App Inventor** is an intuitive, visual programming environment that allows everyone – even children – to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to move from technology consumption to technology creation.

****Blocks-based coding programs inspire intellectual and creative empowerment. MIT App Inventor goes beyond this to provide real empowerment for kids to make a difference -- a way to achieve social impact of immeasurable value to their communities.

 MIT App Inventor is changing the way the world creates apps and the way that kids learn about computing.



4.SERIAL MONITOR:



This window is called the Serial Monitor and it is part of the Arduino IDE software. Its job is to allow you to both send messages from your computer to an Arduino board (over USB) and also to receive messages from the Arduino.  
  
The message “Enter LED Number 0 to 9 or 'x' to clear” has been sent by the Arduino, and it is telling us what commands we can send to the Arduino which is either to send the 'x' (to turn all the LEDs off) or the number of the LED you want to turn on (where 0 is the bottom LED, 1 is the next one up right up to 7 for the top LED).

Try typing the following commands, into the top area of the Serial Monitor that is level with the 'Send' button. Press 'Send', after typingeach of these characters: x 0 3 5

Typing x, will have no effect, if the LEDs are already all off, but as you enter each number, the corresponding LED should light and you will get a confirmation message from the Arduino board, so that the Serial Monitor will appear as shown above.

5.OLED DISPLAY:





The Adafruit library for the OLED display comes with several functions to write text. The Adafruit library for the OLED display comes with several functions to write text. You can display 128×64 bitmap monocolor images on the OLED display.

It can also displays temperature and humidity readings on the OLED display. We’ll get temperature and humidity using the [DHT11 temperature and humidity sensor](https://makeradvisor.com/tools/dht11-temperature-humidity-sensor/).

ARDUINO CODE:

#include "DHT.h"

#define DHTPIN D4

#include "DHT.h"

#define DHTPIN D4

#define DHTTYPE DHT11 // DHT 11

#include <Adafruit\_GFX.h>

#include <Adafruit\_SPITFT.h>

#include <gfxfont.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

const char\* ssid = "Redmi";

const char\* password = "durgarajdd";

#define ORG "2soh8t"

#define DEVICE\_TYPE "sovereigns"

#define DEVICE\_ID "abcd"

#define TOKEN "abcde123"

String command;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

//Serial.println(clientID);

#define OLED\_RESET D5

void drawStr(uint8\_t x, uint8\_t y, char\* str);

/\* Object named display, of the class Adafruit\_SSD1306 \*/

Adafruit\_SSD1306 display(OLED\_RESET);

#if (SSD1306\_LCDHEIGHT != 64)

//#error("Height incorrect, please fix Adafruit\_SSD1306.h!");

#endif

char k[30],s[40];

DHT dht(DHTPIN, DHTTYPE);

WiFiClient wifiClient;

void callback(char\* topic, byte\* payload, unsigned int payloadLength);

PubSubClient client(server, 1883, callback, wifiClient);

// 'aits new cropped', 128x64px

const unsigned char myBitmap1 [] PROGMEM = {

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x01, 0xe3, 0x64, 0xc6, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xe0, 0x0d, 0xe1, 0xf2, 0x2c, 0xd9, 0x80, 0x07, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0x02, 0x79, 0xf0, 0xfb, 0xe7, 0x90, 0x1c, 0xc0, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xf0, 0x19, 0x19, 0x88, 0x00, 0x00, 0x0f, 0x7c, 0x78, 0x0f, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xc0, 0x0f, 0x8c, 0x00, 0x04, 0x80, 0x00, 0x04, 0xc6, 0x03, 0xff, 0xff, 0xff,

0xff, 0xff, 0xfe, 0x03, 0xe4, 0x40, 0x3f, 0xff, 0xff, 0xfc, 0x01, 0x89, 0xf0, 0x7f, 0xff, 0xff,

0xff, 0xff, 0xf8, 0x31, 0xf8, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x12, 0x10, 0x1f, 0xff, 0xff,

0xff, 0xff, 0xe0, 0x9d, 0x80, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xff, 0x03, 0xe7, 0x07, 0xff, 0xff,

0xff, 0xff, 0x80, 0x66, 0x07, 0xff, 0xf0, 0x5f, 0xff, 0xff, 0xff, 0xf0, 0x1a, 0xc1, 0xff, 0xff,

0xff, 0xfe, 0x0e, 0x30, 0x3f, 0xff, 0xe2, 0x2f, 0xff, 0xff, 0xff, 0xfc, 0x07, 0x80, 0x7f, 0xff,

0xff, 0xf8, 0x18, 0xc1, 0xff, 0xff, 0xeb, 0x33, 0xff, 0xff, 0xff, 0xff, 0x86, 0x7c, 0x1f, 0xff,

0xff, 0xf0, 0x07, 0x87, 0xff, 0xff, 0xe0, 0x01, 0xff, 0xff, 0xff, 0xff, 0xe1, 0x78, 0x0f, 0xff,

0xff, 0xe1, 0xfc, 0x1f, 0xff, 0xff, 0xf0, 0x0c, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x33, 0x87, 0xff,

0xff, 0x80, 0x70, 0x7f, 0xff, 0xff, 0xf0, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xfe, 0x0c, 0x01, 0xff,

0xff, 0x0f, 0x20, 0xff, 0xff, 0x0f, 0xf0, 0x33, 0x3f, 0xff, 0xff, 0xf8, 0x07, 0x8c, 0x00, 0xff,

0xfe, 0x03, 0xc3, 0xff, 0xff, 0x83, 0xfc, 0xf1, 0x1f, 0xff, 0xff, 0xdf, 0xff, 0xc0, 0x08, 0x7f,

0xfc, 0x3f, 0x87, 0xff, 0xff, 0x40, 0x7c, 0x3c, 0x07, 0xff, 0xff, 0xbf, 0xff, 0xe0, 0x1c, 0x3f,

0xf8, 0x03, 0x0f, 0xff, 0xf9, 0xc0, 0x1c, 0x0f, 0x10, 0x3f, 0xff, 0x0f, 0xff, 0xf8, 0xe2, 0x1f,

0xf0, 0xfe, 0x1f, 0xff, 0xfe, 0x70, 0xc1, 0x00, 0x25, 0x1f, 0xff, 0x0f, 0xff, 0xfc, 0x07, 0x0f,

0xf0, 0x78, 0x3f, 0xff, 0xff, 0x9e, 0xd0, 0x00, 0x1b, 0x1f, 0x7f, 0xff, 0xff, 0xfe, 0x3d, 0x8f,

0xe0, 0x18, 0x7f, 0xff, 0xff, 0xf1, 0xfc, 0x1f, 0xc6, 0x4c, 0x0f, 0xff, 0xff, 0xfe, 0x0f, 0x87,

0xc0, 0xf8, 0xff, 0xff, 0xff, 0xff, 0xf3, 0x80, 0xf8, 0x4c, 0x93, 0xff, 0xff, 0xff, 0x10, 0xc3,

0xc7, 0xc1, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xe0, 0x00, 0x5f, 0xbb, 0xff, 0xff, 0xff, 0x8e, 0x03,

0x80, 0xe1, 0xff, 0xff, 0xff, 0xfe, 0x86, 0x11, 0xc0, 0x5f, 0xff, 0xff, 0xff, 0xff, 0x8d, 0xe1,

0x87, 0xe3, 0xff, 0xff, 0xff, 0xfe, 0xef, 0xa7, 0x80, 0x2f, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x01,

0x83, 0xc3, 0xff, 0xff, 0xff, 0xfc, 0x8e, 0x09, 0xe3, 0x2f, 0x7f, 0xff, 0xff, 0xff, 0xc3, 0x71,

0x8f, 0xc7, 0xff, 0xff, 0xff, 0xff, 0x3f, 0x0f, 0x88, 0x12, 0x7f, 0xff, 0xff, 0xff, 0xe6, 0xd1,

0x00, 0x47, 0xff, 0xff, 0xff, 0xfd, 0xfe, 0x00, 0xe0, 0x01, 0xff, 0xff, 0xff, 0xff, 0xe1, 0x00,

0x1f, 0x87, 0xff, 0xff, 0xff, 0xfd, 0x20, 0x17, 0xe4, 0x03, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x18,

0x07, 0x87, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x03, 0x00, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xe3, 0xf8,

0x00, 0x47, 0xff, 0xff, 0xff, 0xff, 0xc8, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00,

0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0x8b, 0x30, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00,

0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0x86, 0x09, 0x59, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00,

0x83, 0x07, 0xff, 0xff, 0xff, 0xff, 0x88, 0xc9, 0x09, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0xc1,

0x87, 0x83, 0xff, 0xff, 0xff, 0xff, 0x04, 0x19, 0x11, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc3, 0xe1,

0x84, 0x83, 0xff, 0xff, 0xff, 0xff, 0x0d, 0xd9, 0x09, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc1, 0x61,

0x80, 0x01, 0xff, 0xff, 0xff, 0xfe, 0x0e, 0xf9, 0x09, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x01,

0xc0, 0x09, 0xff, 0xff, 0xff, 0xff, 0x07, 0xf1, 0x18, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x03,

0xc0, 0x70, 0xff, 0xff, 0xff, 0xff, 0xc3, 0xf0, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1a, 0x43,

0xe1, 0xfc, 0x7f, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xfe, 0x3e, 0x87,

0xf0, 0x02, 0x3f, 0xff, 0xff, 0xff, 0xf8, 0x02, 0x08, 0x07, 0xff, 0xff, 0xff, 0xfe, 0x63, 0x8f,

0xf0, 0x7c, 0x3f, 0xff, 0xff, 0xff, 0xfc, 0x09, 0x18, 0x1f, 0xff, 0xff, 0xff, 0xfc, 0x1e, 0x0f,

0xf8, 0x07, 0x8f, 0xff, 0xff, 0xff, 0xfd, 0x59, 0xb0, 0x7f, 0xff, 0xff, 0xff, 0xf8, 0xf0, 0x1f,

0xfc, 0x18, 0xc7, 0xff, 0xff, 0xff, 0xfe, 0x88, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xe3, 0x1c, 0x3f,

0xfe, 0x0f, 0xa3, 0xff, 0xff, 0xff, 0xfe, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc7, 0x00, 0x7f,

0xff, 0x00, 0x61, 0xff, 0xff, 0xff, 0xfc, 0x23, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0xe0, 0xff,

0xff, 0x81, 0xf8, 0x7f, 0xff, 0xff, 0xe0, 0x1b, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x3c, 0x01, 0xff,

0xff, 0xe0, 0x36, 0x1f, 0xff, 0xff, 0xc1, 0x81, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x9f, 0x83, 0xff,

0xff, 0xf0, 0x60, 0xc7, 0xff, 0xff, 0xeb, 0xe4, 0xff, 0xff, 0xff, 0xff, 0xe0, 0xcc, 0x0f, 0xff,

0xff, 0xf8, 0x1e, 0xc1, 0xff, 0xff, 0xe7, 0xe8, 0xff, 0xff, 0xff, 0xff, 0x87, 0xc0, 0x1f, 0xff,

0xff, 0xfe, 0x01, 0x9c, 0x3f, 0xff, 0xfe, 0x7b, 0xbf, 0xff, 0xff, 0xfe, 0x18, 0x30, 0x7f, 0xff,

0xff, 0xff, 0x81, 0x39, 0x0f, 0xff, 0xf8, 0x3c, 0x0f, 0xff, 0xff, 0xf1, 0x06, 0x01, 0xff, 0xff,

0xff, 0xff, 0xe0, 0x4e, 0x00, 0xff, 0xff, 0xff, 0xd0, 0xbf, 0xff, 0x08, 0xce, 0x07, 0xff, 0xff,

0xff, 0xff, 0xf8, 0x08, 0x1e, 0x0f, 0xff, 0xff, 0xfa, 0x7f, 0xf0, 0x66, 0x60, 0x1f, 0xff, 0xff,

0xff, 0xff, 0xfe, 0x00, 0x2d, 0x10, 0x3f, 0xff, 0xff, 0xfe, 0x01, 0xb3, 0xc0, 0x7f, 0xff, 0xff,

0xff, 0xff, 0xff, 0xc0, 0x36, 0x78, 0xc0, 0x0d, 0xc0, 0x00, 0xc0, 0xc0, 0x03, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xf0, 0x01, 0xf9, 0x9e, 0x26, 0x7d, 0xf9, 0xe0, 0x40, 0x0f, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xfe, 0x00, 0x1b, 0x62, 0x4f, 0x19, 0xf1, 0xf8, 0x00, 0x7f, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x33, 0x9f, 0x18, 0xdd, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x10, 0x88, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff

};

// 'cocola image new cropped', 128x64px

const unsigned char myBitmap2 [] PROGMEM = {

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0x36, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbf, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x71, 0xff, 0xff, 0xff, 0xff, 0xff, 0x28, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x00, 0xf1, 0xff, 0xff, 0xff, 0xff, 0xff, 0x60, 0x20, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x0e, 0xa9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xa0, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x0c, 0x08, 0x3b, 0xf7, 0xef, 0xff, 0xfc, 0x00, 0x80, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x00, 0x3c, 0x00, 0x33, 0x3f, 0xcf, 0x7f, 0xf8, 0x00, 0x80, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x03, 0xfc, 0x00, 0x38, 0x7f, 0x8c, 0xff, 0xfc, 0x00, 0x80, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x1f, 0xfd, 0x80, 0x1f, 0xe7, 0xef, 0x77, 0xfc, 0x00, 0x80, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0x7f, 0xfd, 0xe0, 0x00, 0x00, 0x00, 0x00, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x00, 0xff, 0xfd, 0xfc, 0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x08, 0xff, 0xfd, 0xff, 0x80, 0xff, 0xf7, 0xff, 0x01, 0xff, 0x00, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x01, 0xff, 0xfd, 0xff, 0xfc, 0x00, 0x10, 0x00, 0x7f, 0xff, 0x88, 0x60, 0x00, 0x0f,

0xf0, 0x00, 0x03, 0xff, 0xfc, 0x7f, 0xff, 0xff, 0x47, 0xff, 0x07, 0x34, 0x3f, 0xff, 0x00, 0x0f,

0xf0, 0x00, 0x0f, 0xff, 0xf4, 0x07, 0xff, 0xff, 0xfe, 0x00, 0x07, 0xf0, 0x7f, 0xfe, 0x00, 0x0f,

0xf0, 0x00, 0x0f, 0xff, 0xec, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x68, 0x7f, 0xfc, 0x00, 0x0f,

0xf0, 0x01, 0xdf, 0xdf, 0xf4, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x40, 0x6f, 0xf8, 0x00, 0x0f,

0xf0, 0x07, 0xff, 0xc2, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x79, 0x00, 0x00, 0x0f,

0xf0, 0x01, 0xff, 0xa8, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x7a, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0xff, 0x80, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x74, 0x00, 0x00, 0x0f,

0xf0, 0x03, 0xff, 0x80, 0x03, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x2c, 0x00, 0x00, 0x0f,

0xf0, 0x01, 0xff, 0x80, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x00, 0x0f,

0xf0, 0x03, 0xff, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x07, 0xff, 0xf0, 0xc8, 0x00, 0x00, 0x0f,

0xf0, 0x01, 0xff, 0x00, 0x01, 0x80, 0x3f, 0xff, 0xfe, 0x1f, 0xff, 0xf0, 0xc0, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0xff, 0x00, 0x01, 0xff, 0xff, 0xff, 0xfe, 0x1d, 0xfe, 0xf0, 0xc0, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0xff, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe3, 0xb0, 0x80, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0xbf, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf1, 0xf0, 0xc0, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0x3f, 0x30, 0xe1, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0xd0, 0x4e, 0x80, 0x00, 0x0f,

0xf0, 0x00, 0x1f, 0x00, 0x01, 0xff, 0xff, 0xfb, 0xfd, 0xff, 0xfe, 0x70, 0x5e, 0xc0, 0x00, 0x0f,

0xf0, 0x00, 0x0e, 0x00, 0x00, 0xff, 0xff, 0xff, 0xf7, 0xe7, 0xfe, 0x30, 0x50, 0x00, 0x00, 0x0f,

0xf0, 0x00, 0xfe, 0x00, 0x00, 0xf0, 0x00, 0x00, 0x3a, 0xc0, 0x10, 0x10, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x80, 0x03, 0x00, 0x39, 0xe0, 0x00, 0x10, 0x40, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x26, 0xe0, 0x00, 0x10, 0x40, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x08, 0x00, 0x40, 0x02, 0x00, 0x00, 0x3c, 0xe0, 0x00, 0x10, 0x20, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x18, 0xc1, 0xe0, 0x00, 0x3f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x04, 0xc0, 0x10, 0x00, 0x30, 0x40, 0x10, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x04, 0x00, 0x08, 0x00, 0x00, 0x3f, 0xe0, 0x10, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x40, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00, 0x80, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x07,

0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x0f,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff

};

// 'galaxy new cropped', 128x64px

const unsigned char myBitmap3 [] PROGMEM = {

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x80, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xc0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xc0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x03, 0x83, 0x0f, 0x06, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x06, 0x5f, 0x7c, 0xa5, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x04, 0xff, 0xf6, 0xa6, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x06, 0xef, 0x94, 0xe3, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x03, 0xff, 0xf6, 0xe5, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x03, 0x9f, 0x7f, 0x47, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x0f, 0x40, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x01, 0xe0, 0x00, 0x40, 0x00, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x03, 0xf8, 0x3c, 0x02, 0x01, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x0f, 0xf0, 0x18, 0x03, 0x83, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0xe3, 0xff, 0xff, 0xdf, 0xf7, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0xe3, 0xdf, 0xfd, 0xfb, 0xbf, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x20, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0xe0, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x08, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x31, 0x1e, 0x80, 0x3e, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xfe, 0xf0,

0x00, 0x08, 0x10, 0xfe, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xff, 0xff,

0x40, 0x0b, 0xff, 0xff, 0xff, 0xdc, 0xa0, 0x7f, 0xcf, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xff, 0xff,

0x0b, 0xff, 0xff, 0xff, 0x00, 0x07, 0xfd, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xfe, 0x7d, 0xff, 0xff,

0xfb, 0xf8, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xe0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,

0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0d, 0x80, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x30, 0x07, 0x80, 0x00, 0x04, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x07, 0xf0, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xc0, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x07, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x07, 0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0x81, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xfc, 0x00, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0x80, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xfc, 0x00, 0x7f, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x03, 0xf8, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x1f, 0x80, 0x00, 0x00

};

// 'smart1 new cropped', 128x64px

const unsigned char myBitmap4 [] PROGMEM = {

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xc0, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x3c, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf0, 0x3c, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x3c, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x18, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x18, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x18, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x08, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x88, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x40, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xe0, 0x00,

0x00, 0x00, 0xff, 0xc0, 0xfe, 0x3f, 0x80, 0xfc, 0x00, 0xff, 0xe0, 0x1e, 0x8f, 0xdf, 0xfa, 0x00,

0x00, 0x0f, 0xff, 0xf8, 0xff, 0xff, 0xe7, 0xff, 0x0f, 0xff, 0xfe, 0x3f, 0xff, 0xff, 0xff, 0x00,

0x00, 0x1f, 0xff, 0xfc, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xff, 0xff, 0x3f, 0xff, 0xff, 0xff, 0x00,

0x00, 0x1f, 0xc1, 0x00, 0xff, 0xff, 0xff, 0xff, 0xdf, 0xe0, 0xff, 0x3f, 0xff, 0x7f, 0xff, 0x00,

0x00, 0x1f, 0xff, 0xe0, 0xff, 0x07, 0xf8, 0x3f, 0xc0, 0x1f, 0xff, 0x3f, 0xe0, 0x1f, 0xe0, 0x00,

0x00, 0x0f, 0xff, 0xfc, 0xff, 0x07, 0xf8, 0x3f, 0xcf, 0xff, 0xff, 0x3f, 0xc0, 0x1f, 0xe0, 0x00,

0x00, 0x00, 0xff, 0xfe, 0xff, 0x07, 0xf8, 0x3f, 0xdf, 0xfc, 0x7f, 0x3f, 0xc0, 0x1f, 0xe0, 0x00,

0x00, 0x0f, 0xc0, 0xfe, 0xff, 0x07, 0xf8, 0x3f, 0xff, 0xc0, 0xff, 0x3f, 0xc0, 0x1f, 0xe0, 0x00,

0x00, 0x3f, 0xff, 0xfe, 0xff, 0x07, 0xf8, 0x3f, 0xff, 0xff, 0xff, 0x3f, 0xc0, 0x1f, 0xfe, 0x00,

0x00, 0x1f, 0xff, 0xfc, 0xff, 0x07, 0xf8, 0x3f, 0xdf, 0xff, 0xff, 0xbf, 0xc0, 0x0f, 0xff, 0x00,

0x00, 0x03, 0xff, 0xe0, 0xff, 0x07, 0xf8, 0x3f, 0xc7, 0xfe, 0x3f, 0xbf, 0xc0, 0x03, 0xfe, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x18, 0x16, 0x0a, 0x0e, 0x1c, 0x18, 0x30, 0xc0, 0x83, 0xc3, 0x60, 0x03, 0xc1, 0x83, 0x00,

0x00, 0x3c, 0x1e, 0x0c, 0x0e, 0x1c, 0x08, 0x30, 0xe0, 0x82, 0xc1, 0xe0, 0x82, 0xc1, 0x83, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

};

void setup() {

display.clearDisplay();

Serial.begin(9600);

Serial.println("DHTxx test!");

dht.begin();

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); /\* Initialize display with address 0x3C \*/

display.clearDisplay();

display.setTextSize(1); /\* Select font size of text. Increases with size of argument. \*/

display.setTextColor(WHITE); /\* Color of text\*/

Serial.begin(115200);

Serial.println();

// pinMode(D1,OUTPUT);

wifiConnect();

mqttConnect();

}

void loop() {

{

display.clearDisplay();

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

int h = dht.readHumidity();

// Read temperature as Celsius (the default)

int t = dht.readTemperature();

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

}

if (!client.loop()) {

mqttConnect();

}

delay(100);

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void drawStr(uint8\_t x, uint8\_t y, char\* str){

display.setCursor(x, y); // Set x,y coordinates

display.println(str);

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) {

Serial.print("callback invoked for topic: "); Serial.println(topic);

for (int i = 0; i < payloadLength; i++) {

//Serial.println((char)payload[i]);

command += (char)payload[i];

}

Serial.println(command);

int h = dht.readHumidity();

int t = dht.readTemperature();

if(t >20)

{

drawStr(0,20,"temperature is high");

display.display();

delay(5000);

display.clearDisplay();

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.println(" \*C ");

sprintf(k, "Humidity: %d", h);

sprintf(s, "Temperature: %d", t);

drawStr(0, 0,k);

drawStr(0,10,s);

display.display();

delay(5000);

display.clearDisplay();

}

else if(t<20)

{ Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.println(" \*C ");

sprintf(k, "Humidity: %d", h);

sprintf(s, "Temperature: %d", t);

drawStr(0, 0,k);

drawStr(0,10,s);

display.display();

}

display.clearDisplay();

if(command == "A1"){

display.clearDisplay();

display.drawBitmap(0, 0, myBitmap1, 128, 64, BLACK, WHITE);

display.display();

delay(2000);

//display.clearDisplay();

}

if(command == "A2"){

display.clearDisplay();

display.drawBitmap(0, 0, myBitmap2, 128, 64, BLACK, WHITE);

display.display();

delay(2000);

// display.clearDisplay();

}

if(command == "A3"){

display.clearDisplay();

display.drawBitmap(0, 0, myBitmap3, 128, 64, BLACK, WHITE);

display.display();

delay(2000);

//display.clearDisplay();

}

if(command == "A4"){

display.clearDisplay();

display.drawBitmap(0, 0, myBitmap4, 128, 64, BLACK, WHITE);

display.display();

delay(2000);

//display.clearDisplay();

}

/\*if(command == "ALLON"){

digitalWrite(D1,HIGH);

digitalWrite(D2,LOW);

Serial.println("Light is Switched ON");

}

else if(command == "ALLOFF"){

digitalWrite(D1,LOW);

digitalWrite(D2,HIGH);

Serial.println("Light is Switched OFF");

} \*/

// Length (with one extra character for the null terminator)

int str\_len = command.length() + 1;

// Prepare the character array (the buffer)

char c[str\_len];

// Copy it over

command.toCharArray(c, str\_len);

//sprintf(c,"%s",command);

Serial.print("c : ");

Serial.println(c);

display.clearDisplay();

drawStr(0,10,c);

display.display();

command ="";

}

…………………………………………………………………………………………………………...