

Assignment.

Manzi Uniyal
19EE10D29

Q1. Advantage of 32-bit Microcontroller over 8 bit.

- ① Superior processing speed: 8bit MC controller runs at 8 MHz to 16MHz clock speed.
32 bit MC controller can be clocked upto hundreds of MHz. (ESP32 → 240MHz)
- ② More RAM (8times): 32bit MC controller more efficient with large size of data and provides higher accuracy in small data.
- ③ Wider databus, few instruction cycles to execute function:

Q2. Specifications of ESP32.

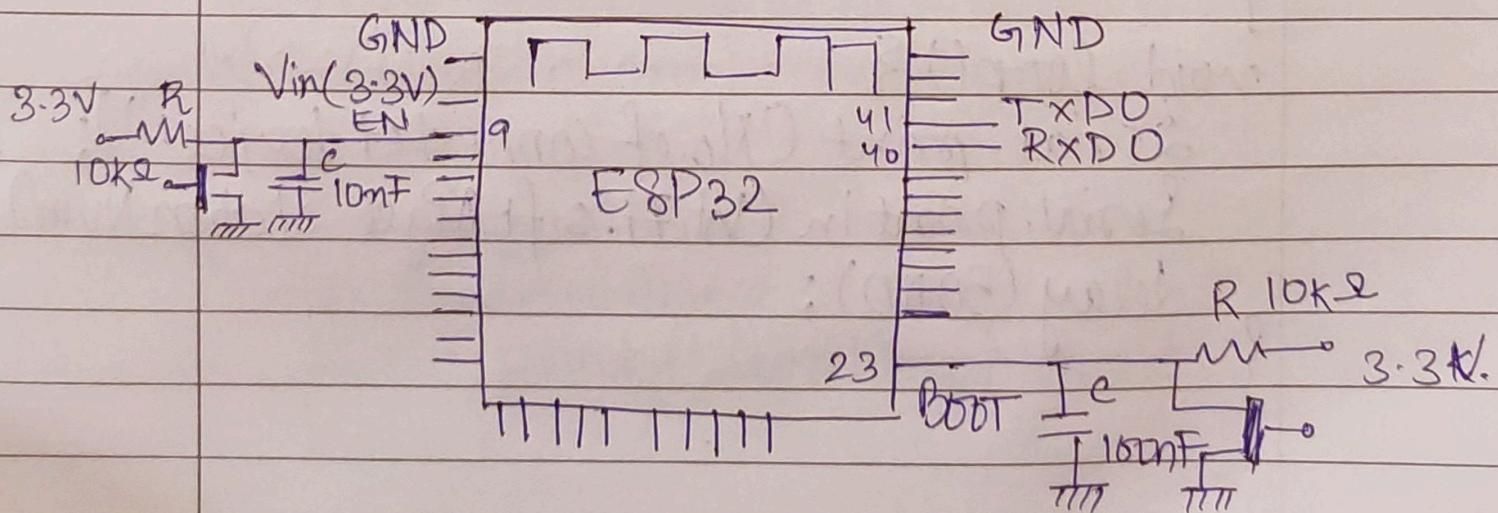
- Single/dual core 32bit LX6 micro processor with Clock freq upto 240MHz.
- 520kB of SRAM, 448kB of ROM, 16kB of RTC SRAM.
- 802.11 b/g/n Wi-Fi connectivity with speeds upto 150Mbps.
- Supports both classic Bluetooth v4.2 and BLE specifications.
- 34 programmable GPIOs.
- Upto 18 channels of 12bit SAR ADC and 2 channels of 8bit DAC.

- Serial connectivity of 4xSPI, 2xI2S, 3xUART
- Ethernet MAC for physical CAN communication
- 1 Host controller for SD/SDIO/MMC and 1 slave controller for SDIO/SPI.
- Motor PWM, upto 16 channels of LED PWM.
- Secure Boot and flash encryption.

Q3.

ESP32 gives multiple programming environment: Some examples are:-

- ① Arduino IDE
- ② PlatformIO IDE (VS code)
- ③ LUA
- ④ Micro Python
- ⑤ Espressif IDE (IoT Development Framework)
- ⑥ JavaScript



Q4. Access Point (AP) for Web Server.

- Connect ESP8266 with PC using OTG cable.
- open Arduino IDE.
- In 'File' section, click 'Preferences'
- ~~Add~~ Add ESP8266 URL for Arduino IDE at 'Additional Boards Manager URLs'
- In 'Tools' section, select 'Board Manager' and install latest version of 'Esp8266'
- In 'Tools' section, select 'Manage Libraries' and install 'Arduino UNO WiFi Dev Ed Lib'.
- code in Arduino IDE:-

~~#include~~

#include <ESP8266 WiFi.h>

void setup() {

Serial.begin(9600);

WiFi.softAP('ESP', '0000 0000');

Serial.print('Access Point IP: ');

Serial.println(WiFi.softAPIP());

}

void loop() {

Serial.print('No. of connected device: ');

Serial.println(WiFi.softAPgetStationNum());

delay(5000);

}

- Select NodeMCU 1.0 (ESP12E Module) under 'ESP8266 Boards(3.0.2)'
- compile and upload to ESP8266
- Once uploaded, access point gets cracked and we connect devices (Phone) to ESP8266 with password.
- Check no. of connected devices and access point IP using Serial Monitor.

Q5.

- Create account at ThingSpeak
- Provided necessary libraries and boards installed, ESP8266 device connected and preferences set, code in Arduino IDE:-

```
#include <WiFi.h>
#include <ThingSpeak.h>
#define CHANNEL_ID 1668369
#define CHANNEL_API_KEY 'D5B7RJAPABR9CA
WiFiClient client;
void setup() {
    Serial.begin(9600);
    WiFi.disconnect();
    Serial.print('Searching Network... ');
    WiFi.begin('moto g6 plus 1137',
              '187654321');
```

```
while (!WiFi.status() == WL_CONNECTED)
{
    Serial.println("...!");
    delay(300);
}

Serial.println("Connected Successfully!");
Serial.print("Your IP: ");
Serial.println(WiFi.localIP());
ThinkSpeak.begin(client);
}

void loop()
{
    int i;
    i = 0;
    while (i < 100)
    {
        ThinkSpeak.writeField("CHANNEL-ID",
            i, "CHANNEL-API-KEY");
        delay(16000);
        i++;
    }
}
```

- Compile and upload code to ESP8266. If we open serial monitor and press Reset pin of ESP8266. WiFi is connected successfully. It will show IP address.

Note:

— / —

Get the Channel id and API key, first in ThingSpeak, we have to create a New Channel, name it, select number of required fields, name fields. On creation of channel, channel id and API key will be available. From 'Add Widgets' we should add 'Numerical Display' to show numerical value sent.