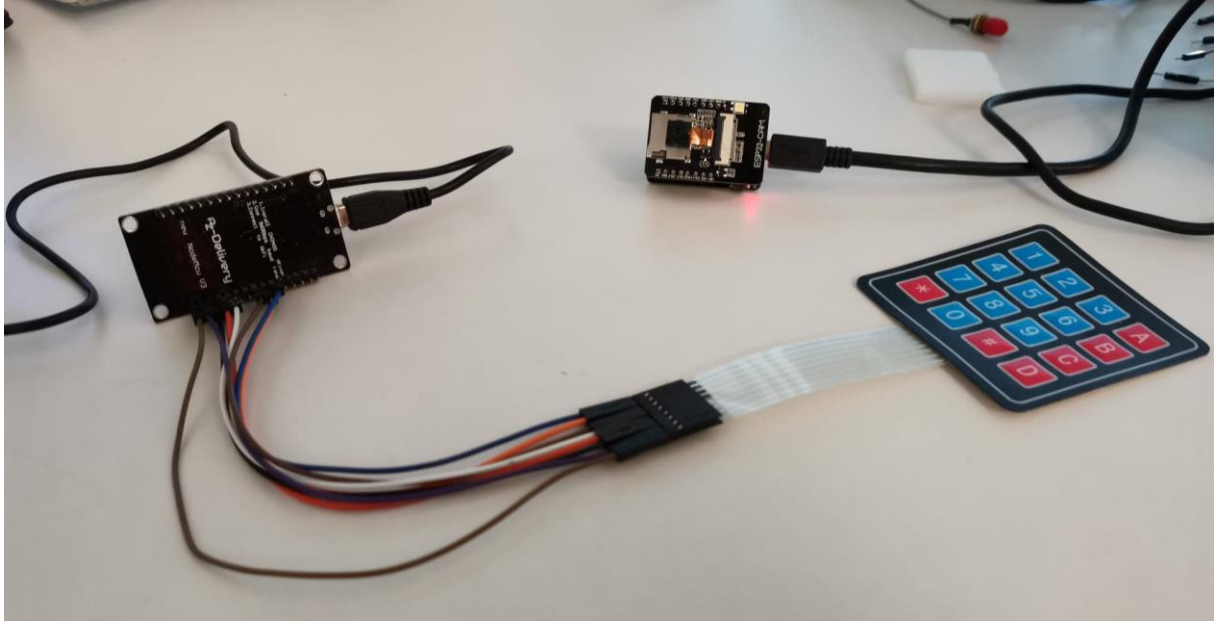


Smart doorbell



Things used in this project

Hardware component

- ESP8266
- ESP32 with camera
- Keypad 4x4
- Jumper Wires 8 male-femelle
- 2 micro-USB cables

Software apps

- Arduino IDE
- MQTT
- Scaleway

Story

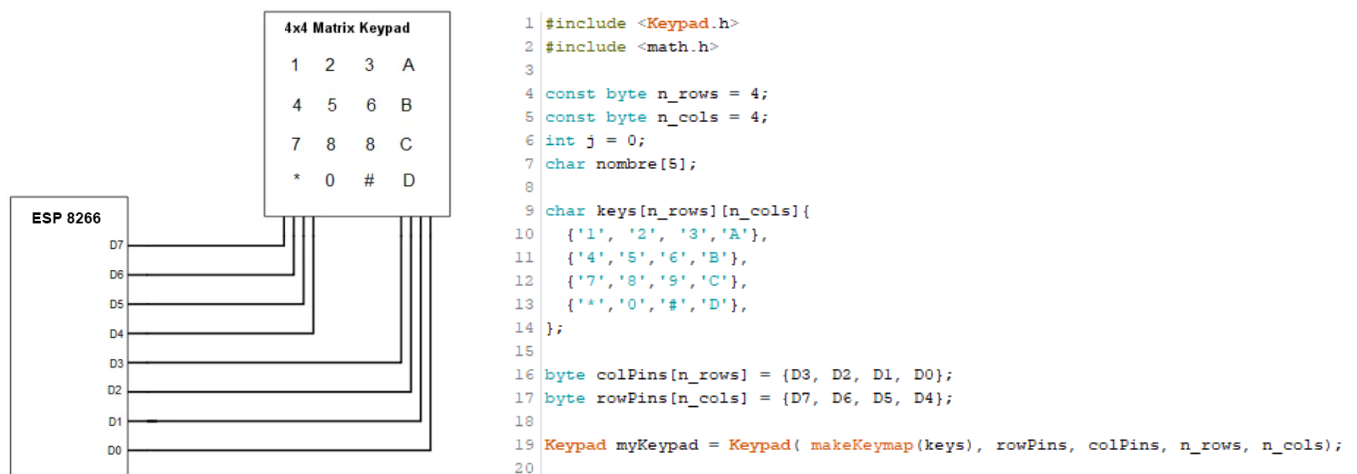
We wanted to create a smart doorbell to provide noise pollution and to save time. Our goal was to be able to ring at specific door number like in a building. When someone rings at your door, the smart doorbell will take a picture and stock this picture in a cloud and maybe be able to recognize who is ringing by an Artificial intelligence if it is someone that comes often.

Working principle

Step 1:

ESP8266 is used to control our connected bell, it serves here as a computer which manages the information. It is link to our keypad 4x4 which is a simulation of a building's intercom. The first step for us was to be able to read which number were pressed on the keypad.

For this one we must connect our keypad to our esp8266 like the architecture below:



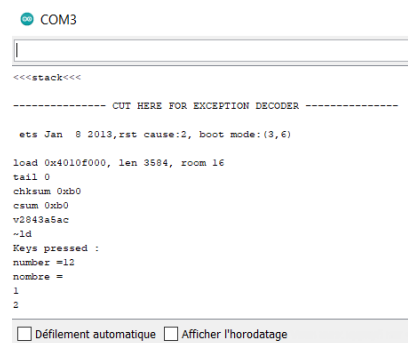
Architecture and initialization code of the keypad

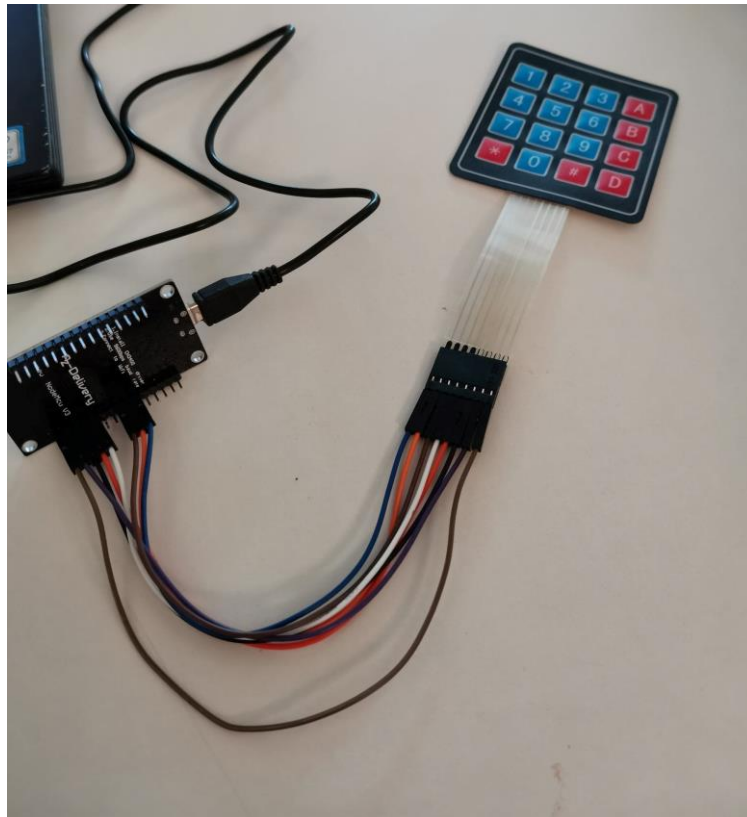
Once the initialization is effective, we want to read the number pressed by the user on the Arduino console, so we use a “for loop” that allows us to see which number we want to call.

```

23 void setup() {
24   // put your setup code here, to run once:
25   Serial.begin(115200);
26 }
27
28 void loop() {
29   // put your main code here, to run repeatedly:
30
31   char myKey[4];
32
33   for (int i = 0; i < 4; i++){
34     char tmp = myKeypad.waitForKey();
35     int number = 0;
36
37     if(tmp != 'A'){
38       myKey[i] = tmp;
39     }
40     else{
41       Serial.println("Keys pressed :");
42       int lengthChar = i;
43       char nombre[lengthChar];
44       for (i = 0; i < lengthChar; i++){
45         number = number + (myKey[i] - 48) * pow(10, lengthChar - i - 1); //Because it's in Ascii code
46         nombre[i] = myKey[i];
47       }
48       Serial.print("number =");
49       Serial.println(number);
50       Serial.println("nombre =");
51       for (i = 0; i < lengthChar; i++){
52         Serial.println(nombre[i]);
53       }
54     }
55   }
56 }
57
58 }
59
60 }

```





Keypad and esp 8266

Now we have our keypad set up, now we want to connect and initialize this with MQTT and a Wi-Fi network. For this we created a Hub and three devices on Scaleway which allows us to communicate between our two ESP. To set up MQTT, you can use the code in copy at the end.

Once we established the connection, we send a payload named “Smile” to MQTT.

```
void sendmqtt() {
    Serial.println("Sending payload");
    client.publish("SMILE", "SMILE"); //Topic name
    Serial.println("Payload sent!");
    client.subscribe(topic_PHOTO);
}
```

Thanks to this, and through MQTT, we send a second payload to our ESP 32-CAM which is the condition for the ESP 32 to take a photo

```
void callback(String topic, byte* message, unsigned int length) {
    String messageTemp;
    Serial.println(topic);
    for (int i = 0; i < length; i++) {
        messageTemp += (char)message[i];
    }
    if (topic == topic_PHOTO) {
        take_picture();
    }
    if (topic == topic_FLASH) {
        set_flash();
    }
}
```

Once the photo is taken, we stock her in a cloud by scaleway through a bucket and after we just must download the picture and change the extension by a .jpeg or .png.

Here is an example of a photo taken by our device.



Credits

- Louis Moreau : <https://github.com/luisomoreau>
- ESP8266: Interfacing with a 4x4 Matrix Keypad by Antepher on techtutorialsx