//Bibliotecas utilizadas

#include "ESPino32CAM.h"

#include "ESPino32CAM\_QRCode.h"

ESPino32CAM cam;   //Objeto para captura de imagem

ESPino32QRCode qr; //Objeto para decoficação da imagem

//Define os pinos da câmera

#define PWDN\_GPIO\_NUM     32

#define RESET\_GPIO\_NUM    -1

#define XCLK\_GPIO\_NUM      0

#define SIOD\_GPIO\_NUM     26

#define SIOC\_GPIO\_NUM     27

#define Y9\_GPIO\_NUM       35

#define Y8\_GPIO\_NUM       34

#define Y7\_GPIO\_NUM       39

#define Y6\_GPIO\_NUM       36

#define Y5\_GPIO\_NUM       21

#define Y4\_GPIO\_NUM       19

#define Y3\_GPIO\_NUM       18

#define Y2\_GPIO\_NUM        5

#define VSYNC\_GPIO\_NUM    25

#define HREF\_GPIO\_NUM     23

#define PCLK\_GPIO\_NUM     22

#define flash 4

//Variável para limitar o print no monitor serial se caso o QR Code ainda estiver na frente da câmera

int trava;

void setup() {

  Serial.begin(115200);

  Serial.println("Leitor de QR Code");

  //Define pino do flash

  pinMode(flash,OUTPUT);

  digitalWrite(flash,LOW);//Desliga o flash

  //Configura os pinos da câmera

  camera\_config\_t config;

  config.ledc\_channel = LEDC\_CHANNEL\_0;

  config.ledc\_timer = LEDC\_TIMER\_0;

  config.pin\_d0 = Y2\_GPIO\_NUM;

  config.pin\_d1 = Y3\_GPIO\_NUM;

  config.pin\_d2 = Y4\_GPIO\_NUM;

  config.pin\_d3 = Y5\_GPIO\_NUM;

  config.pin\_d4 = Y6\_GPIO\_NUM;

  config.pin\_d5 = Y7\_GPIO\_NUM;

  config.pin\_d6 = Y8\_GPIO\_NUM;

  config.pin\_d7 = Y9\_GPIO\_NUM;

  config.pin\_xclk = XCLK\_GPIO\_NUM;

  config.pin\_pclk = PCLK\_GPIO\_NUM;

  config.pin\_vsync = VSYNC\_GPIO\_NUM;

  config.pin\_href = HREF\_GPIO\_NUM;

  config.pin\_sscb\_sda = SIOD\_GPIO\_NUM;

  config.pin\_sscb\_scl = SIOC\_GPIO\_NUM;

  config.pin\_pwdn = PWDN\_GPIO\_NUM;

  config.pin\_reset = RESET\_GPIO\_NUM;

  config.xclk\_freq\_hz = 20000000;

  config.pixel\_format = PIXFORMAT\_JPEG;

  config.frame\_size = FRAMESIZE\_VGA;

  config.jpeg\_quality = 4;

  config.fb\_count = 1;

  esp\_err\_t err = esp\_camera\_init(&config); //Inicialização da câmera

  if (err != ESP\_OK) {

    Serial.printf("O início da câmera falhou com erro 0x%x", err);//Informa erro se a câmera não for iniciada corretamente

    delay(1000);

    ESP.restart();//Reinicia o ESP

  }

  //Inicializa o objeto de decodificação

  qr.init(&cam);

  sensor\_t \*s = cam.sensor();

  s->set\_framesize(s, FRAMESIZE\_CIF);

  s->set\_whitebal(s, true);

  Serial.println();

  Serial.println("Aguardando código");

}

void loop()

{

  unsigned long pv\_time  = millis();

  camera\_fb\_t \*fb = cam.capture(); //Captura a imagem

  if (!fb)

  {

    Serial.println("Falha na captura da imagem");

    return;

  }

  dl\_matrix3du\_t \*rgb888, \*rgb565;

  if (cam.jpg2rgb(fb, &rgb888))

  {

    rgb565 = cam.rgb565(rgb888);

  }

  cam.clearMemory(rgb888);

  cam.clearMemory(rgb565);

  dl\_matrix3du\_t \*image\_rgb;

  if (cam.jpg2rgb(fb, &image\_rgb))

  {

    cam.clearMemory(fb);

    qrResoult res = qr.recognition(image\_rgb); //Faz decodificação da imagem contendo os dados

    if (res.status) //Se conseguir decodificar a imagem mostra os dados na tela

    {

      if(trava == 0){

       trava = 1;

       String leitura = "Código QR Lido: " + res.payload;//Variável para mostrar os dados contidos no QR Code

       Serial.println();

       Serial.println(leitura);  //Mostra os dados no monitor serial

      }

    }

    else{ //Se não aguarda receber código

       trava = 0;

       Serial.println();

       Serial.println("Aguardando código");

     }

    }

  cam.clearMemory(image\_rgb); //Apaga imagem para receber uma nova imagem

<https://www.makerhero.com/blog/utilize-esp32-cam-como-leitor-de-qr-code/>