
Relatório de Atividade - ESP32 CAM

Responsável:	Matheus Gois Vieira
Data de Início:	07/01/2020
Data de Entrega:	08/01/2020

1. Objetivo da Atividade

A atividade tem como objetivo testar e validar funcionalidades na Esp32 CAM. Somando a isso, é importante pontuar objetivos específicos:

- Viabilizar conexão da Esp32 Cam com o módulo conversor USB Rs232 TTL, para introduzir o código embarcado no microcontrolador;
- Descrever código embarcado para a captura de imagens;
- Testar funcionalidade de video e imagens da câmera;

2. Descrição da Atividade

Atividade consiste na descrição de do código embarcado, utilizando a plataforma Arduino IDE, que controlará o funcionamento da ESP32 CAM, com isso, foi possível obter imagens através de um IP gerado pela ESP32 CAM, que contém uma interface web que controla os detalhes gráficos do vídeo.

3. Resultados

Para a conexão do Conversor RS232 com a ESP32 CAM foi necessário respeitar o seguinte esquemático produzido no easyEDA mostrado na imagens abaixo.

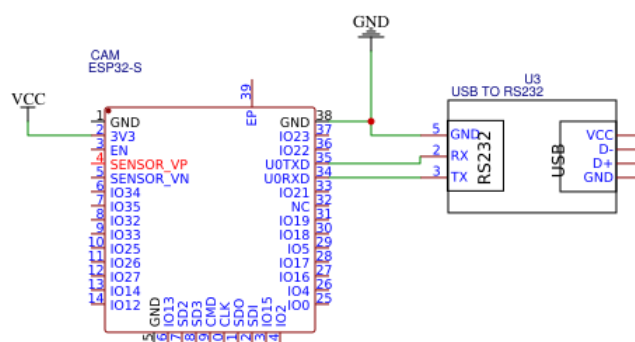


Figura 1: Esquemático produzido no easyEDA, conexão entre o microcontrolador ESP32-CAM e o Conversor USB TTL RS232

É importante se atentar durante a conexão em 2 pontos essenciais para o bom funcionamento, primeiro é alimentação da ESP32 CAM, pois no datasheet da ESPRESSIF está descrito que para o funcionamento de algumas funcionalidades do microcontrolador é preciso alimentá-lo com uma tensão de 5V, apesar de poder alimentá-lo com 3,3V. Além disso, é importante ligar corretamente o TX e o RX, com o UOR e UOT, respectivamente, para que a transmissão de dados seja feita com sucesso.

A construção do código embarcado teve como base o exemplo CamWebServer da esp32, que estão no ARDUINO IDE, mas para o funcionamento da ESP32 CAM, respeitando as conexões feitas anteriormente, é preciso fazer alterações no código. A primeira é habilitar o tipo de microcontrolador utilizado, e a segunda é mudar as configurações de login e senha do wifi local.

```
1 // Selecionando qual modelo de camera
2 // #define CAMERA_MODEL_WROVER_KIT
3 // #define CAMERA_MODEL_ESP_EYE
4 // #define CAMERA_MODEL_M5STACK_PSRAM
5 // #define CAMERA_MODEL_M5STACK_WIDE
6 #define CAMERA_MODEL_AI_THINKER
```

```
1 // Definindo e alterando dados de rede local
2 const char* ssid = "matheusgvieira";
3 const char* password = "gois6523";
```

Com isso, foi possível, gravar videos e adquirir imagens com a esp32 cam.

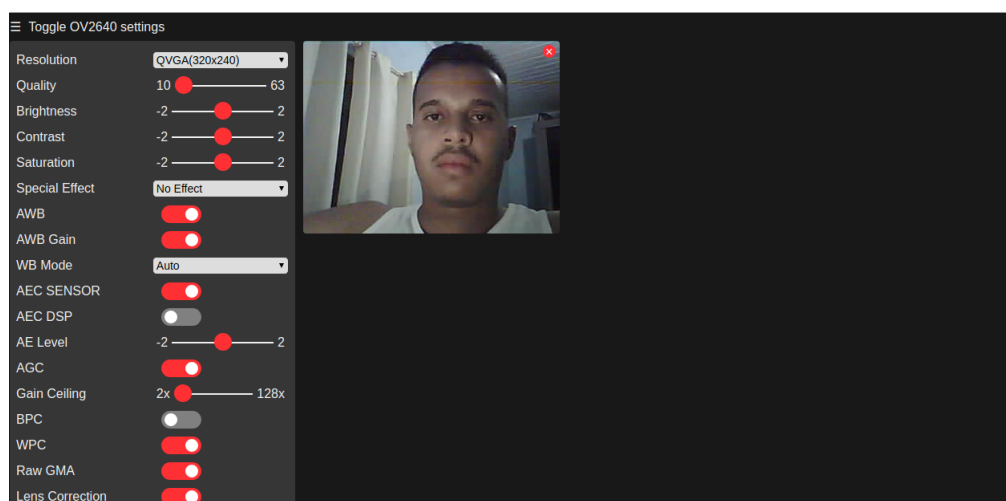


Figura 2: Imagem capturada pela ESP32-CAM com resolução 320x240

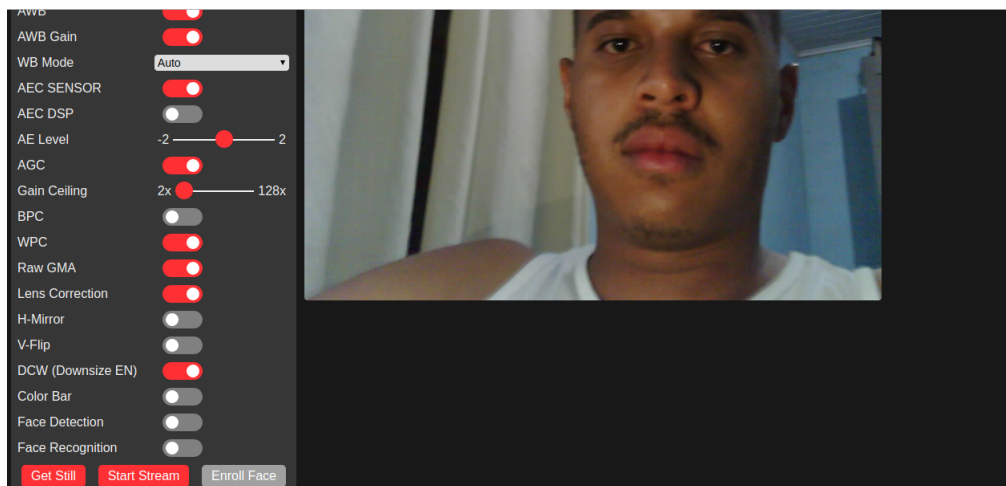


Figura 3: Imagem capturada pela ESP32-CAM com resolução 1600x1200



Figura 4: Imagem capturada e mostrada no monitor serial da IDE do Arduino

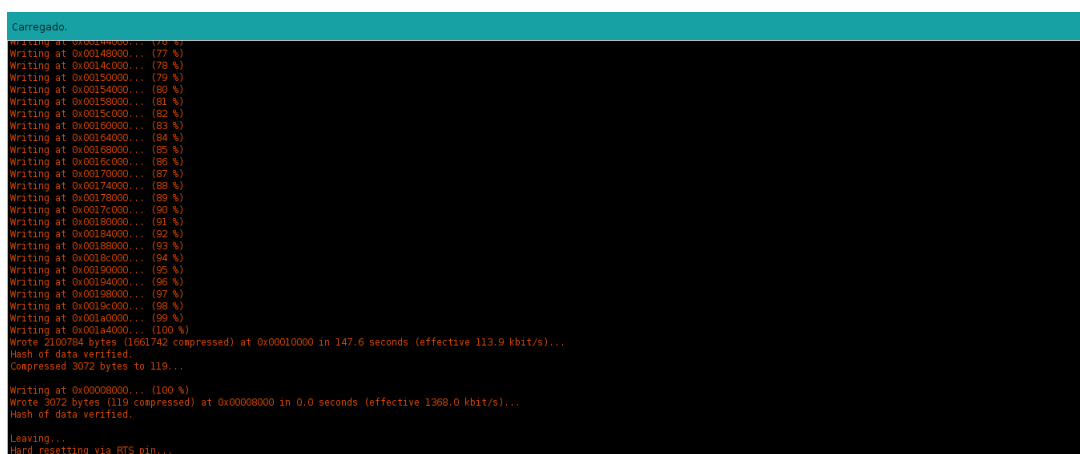


Figura 5: Dados do código carregado na ESP32-CAM

4. Conclusão

Os resultados foram alcançados com o sucesso, dificuldades foram encontradas pelo caminho, como por exemplo o tipo de alimentação para o funcionamento correto do Wifi do microcontrolador, porém foi superado.

A próxima etapa é extrair a imagem para um servidor externo, com o objetivo processa-la e adquirir as informações necessárias para identificar objetos.

5. Referencial Bibliográfico

Espressif. **ESP32-CAM Module**. Disponível em: <<https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf>>. Acesso em 07 de Janeiro de 2020.