

Internet of Things

Uma rede LoRa para envio de imagens

Victor E. Almeida Marco A. Guerra

UNIOESTE

22 de julho de 2022



Conteúdo

- 1 Introdução
- 2 Definições
- 3 Materiais e métodos
 - Algoritmos utilizados
 - Dispositivos utilizados
- 4 Implementação
- 5 Resultados
- 6 Discussão
- 7 Conclusão





Definições I

- **Internet das coisas:** *Internet of things (IoT)*, uma rede que conecta diversas “coisas” a internet, através de software, com o objetivo de trocar informações, tais “coisas” são dispositivos físicos ou lógicos, podem ser sensores, microcontroladores ou até mesmo objetos que nunca imaginamos tais como geladeiras, televisores, entre outros.



Definições II

- **LoRa**

- Atua na camada física;
- Radio frequência;
- Longas distâncias;
- Baixo custo de transmissão;



Algoritmo CRC 16 bits

```
1  uint16_t computeCRC(uint8_t* data_in, uint16_t length) {
2      uint8_t bitbang, j;
3      uint16_t i, crc_calc = INIT;
4      for(i = 0; i < length; i++) {
5          crc_calc ^= (((uint16_t)data_in[i]) & 0x00FF);
6          for(j = 0; j < 8; j++) {
7              bitbang = crc_calc;
8              crc_calc >>= 1;
9              if(bitbang & 1) crc_calc ^= POLY;
10         }
11     }
12     return (crc_calc & 0xFFFF);
13 }
```



Algoritmo Stop and Wait - Sender

```
1 void sender() {  
2     InitCamera();  
3     InitLoRa();  
4     TakePicture(image);  
5     SplitImage(image);  
6     while(image.sendedParts < image.totalParts) {  
7         SendImagePart(image.part());  
8         ReceivePacketCommand(buffer);  
9         if ( buffer .messageType == ACK) {  
10             image.sendedParts++;  
11         }  
12     }  
13 }
```



Algoritmo Stop and Wait - Receiver

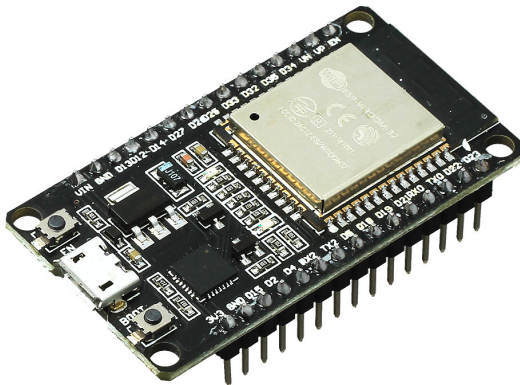
```
1 void reciver () {  
2     InitImage(image);  
3     InitLoRa();  
4     while(true) {  
5         ReceivePacketCommand(buffer);  
6         SavelmageBytes(buffer);  
7         PrepareFrameCommand(); // prepara ACK  
8         SendPacket();  
9         if (image.isComplete()) {  
10             Savelmage(image);  
11         }  
12     }
```



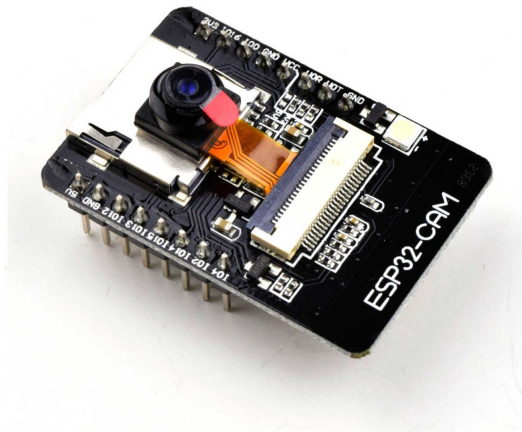
LoRaMESH EndDevice



ESP32



ESP32-CAM



Estruturas de dados enviadas

ID	Command	Payload	CRC
2 bytes	1 byte	1 - 231 bytes	2 bytes

Payload				
Type	ID	Part	Total	Message
1 byte	1 byte	1 byte	1 byte	1 - 227 byte

Payload	
ACK	ID
1 byte	1 byte

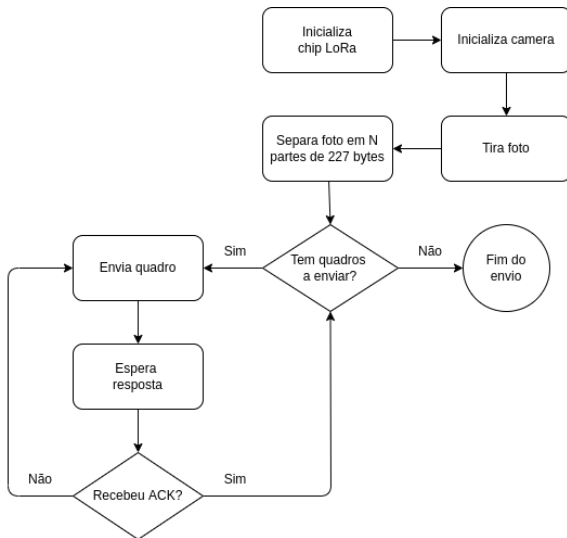


Implementação em código

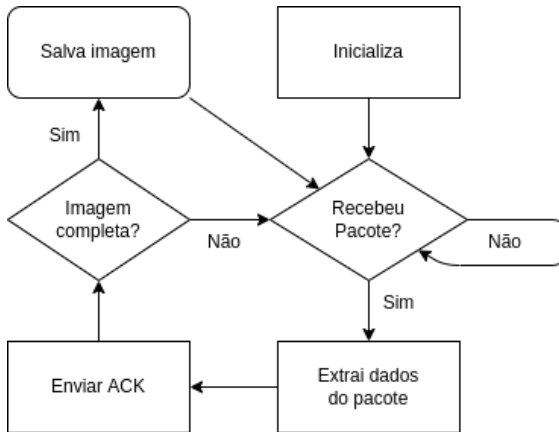
```
1  struct _payload {
2      uint8_t  byte_array [MAX_PAYLOAD_SIZE];
3      uint8_t  size ;
4  };
5
6  struct _fields {
7      uint8_t  type, id, part, last_part ;
8  };
9
10 union ImagePart {
11     _fields  fields ;
12     _payload payload;
13 };
```



Fluxograma Sender



Fluxograma Receiver



Teste de Velocidade de transmissão

- Envia e recebe a resposta em 2 segundos, timeout = 3 segundos;
- Máximo descrito na documentação = 21875 bits por segundo.
- Máximo utilizando stop and wait = $232 * 8 = 1856$ bits por segundo



Testes no tamanho da imagem I

Os testes seguiram os seguintes critérios:

- 3 fotos por resolução escolhendo sempre a mediana.
- Fotos tiradas do mesmo local na mesma posição;
- Imagens em escala de cinsa;



Testes no tamanho da imagem II

Compressão constante em 0

Resolução (pixels)	Tamanho (bytes)
640x480	73260
480x320	39139
400x296	35916
320x240	23510
240x176	14242
176x144	9147

Tabela 2: Mudança de resolução afetando o tamanho da imagem



Testes no tamanho da imagem III

Resolução constante em 480x320

Qualidade (0-63)	Tamanho (bytes)
0	39139
10	8456
20	6371
30	5613
40	5161
50	4842
60	4665
63	4616

Tabela 3: Mudança de qualidade da imagem afetando o tamanho



Mão na massa!!



Agradecimentos

Perguntas?



Obrigado pela atenção

