**PROTOCOLO DE COMUNICAÇÃO ENTRE O PC E O ARDUINO – EMMN**

A comunicação entre o PC e o Arduino é feita via porta USB, a uma taxa de 9600 bps, com 1 bit de parada e sem paridade. O sistema contem três comandos: SET, STATE, POWER. Além disso, existem três tipos de respostas: ACKNOWLEDGED e NOT ACKNOWLEDGED e a resposta ao comando STATE. As respostas ACKNOWLEDGED E NOT ACKNOWLEDGED serão abreviadas aqui por ACK e NACK, respectivamente. Esses comandos serão descritos nos tópicos seguintes. No comando SET, STATE e resposta ao STATE serão enviados um código para verificação de erro na mensagem, o *checksum*.

**COMANDO SET**

Nesse comando, são enviadas para o arduino as posições de referência para as quais ele deve apontar no instante em que receber o comando. Os ângulos de referência são enviados codificados em 16 bits, ou seja, ocupam dois bytes na *string* de mensagem. A codificação é feita de modo que 0° corresponda a 0 em binário, e 360° corresponda a 65535 (2^16 -1). Segue abaixo o modelo da *string* de mensagem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S | A1 | A0 | E1 | E0 | CS |
| 0x57 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

**Comando SET.**

O primeiro byte é o caractere ASCII 'W', que indica o início da mensagem. Os bytes A1 e A0 representam o ângulo de azimute, sendo A1 o byte com os 8 bits mais significativos e A0 o byte com os 8 bits menos significativos. O mesmo vale para os bytes de elevação E1 e E0. O ultimo byte é o *Checksum*, o código para detecção de erro na mensagem. Segue abaixo exemplo da codificação dos ângulos.

**Ex: ângulo de azimute = 235,43°**

Ângulo codificado = 235,43 \* (65535/360) = 42858,069 ≈ 42858

Ângulo codificado em binário = 1010 0111 0110 1010

A1 = 1010 0111 = 167

A0 = 0110 1010 = 106

**COMANDO STATE**

O comando STATE solicita ao arduino o status atual do sistema. Esse comando é enviado através de um único byte, formado pelo caractere ASCII 'S'. Esse comando não envia checksum, porém, a mensagem de resposta ao PC contém.

A resposta a esse comando é uma string com o status atual do sistema com: posição atual da antena; status dos relés; defeito dos inversores; status do controle manual; e status da potência. Além disso, é incluído o byte de *checksum*. Segue abaixo o modelo da string de resposta.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | A1 | A0 | E1 | E0 | C | CS |
| 0x57 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

**Resposta ao comando STATE.**

O primeiro byte representa o início de mensagem 'W'. Os ângulos da posição atual da antena são codificados da mesma forma que no comando SET. O byte C contém informações do status atual do sistema. E o ultimo byte será o checksum, para verificação de erros na mensagem.

**Byte C**

No byte 'C', dos 8 bits que compõe seu byte, são utilizados os 5 menos significativos e o bit mais significativo (MSB). Para cada um dos eventos é enviado o bit 0 caso a situação esteja normalizada e 1 caso algum desses eventos tenha ocorrido. Segue abaixo um modelo da resposta ao comando STATE.

O byte C é codificado da seguinte maneira:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| P | 0 | 0 | KA1 | KA2 | DAZ | DEL | M |

**Formação do caractere de status C.**

Dessa forma, o valor numérico do byte C será:

C= P\*2^7 + KA1\*2^4 + KA2\*2^3+ DAZ\*2^2 +DEL\*2^1 + M\*2^0

Onde:  
P --> Potência (0: desligada, 1: ligada);  
KA1 --> Relé KA1 (0: inativo, 1: ativo);  
KA2 --> Relé KA2 (0: inativo, 1: ativo);  
DAZ --> Defeito no Inversor de Azimute (0: inativo, 1: ativo)  
DEL --> Defeito no Inversor de Elevação (0: inativo, 1: ativo)  
M --> Controle Manual (0: inativo, 1: ativo)

**VERIFICADOR DE ERRO DE BITS – CHECKSUM**

Um código Checksum é utilizado para detecção de erro na transmissão. Esse código consiste em efetuar a operação XOR entre todos os bytes da mensagem e anexar o byte resultante no final da mensagem, antes de sua transmissão. Para verificar a integridade da mensagem no receptor, a mesma operação é feita, mas incluindo o byte de checksum recebido. Um resultado diferente de 0 nesta operação indica que a mensagem está corrompida, e um resultado igual a 0 indica que ou não há erro ou se ocorreu, o erro é indetectável pelo checksum.

**Exemplo de checksum no comando SET:**

Azimute: 115,76° 🡪 Ângulo codificado = 115,76\*65535/360 ≈ 21073

A1 = 0101 0010 = 82

A0 = 0101 0001 = 81

Elevação: 43,23° 🡪 Ângulo codificado = 43,23 \*65535/360 = 7875

E1 = 0001 1110 = 30

E0 = 1100 0011 = 195

Checksum = 87 xor 82 xor 81 xor 30 xor 195

Checksum = 137

Valor numérico da string de comando: {87, 82, 81, 30, 195, 137}

Caso o arduino perceba que o checksum da mensagem chegou errado, será enviado um comando NACK ao PC, que será descrito mais a frente.

**COMANDO POWER**

O comando POWER será enviado pelo PC para ligar ou desligar a potência do sistema. Esse comando será enviado como um único byte, formado pelo caractere 'P'. A função desse comando será alternar o estado do sistema entre ligado e desligado. Esse comando será enviado a cada inicio e fim de rastreio. Não será enviado checksum.

**COMANDO RESPOSTA – ACKNOWLEDGED E NOT ACKNOWLEDGED**

Os comandos ACK e NACK serão codificados por caracteres ASCII 'A'e 'N', respectivamente. Esses comandos serão enviados por ambos os lados, arduino e PC, assim que ambos receberem um comando ou resposta a um comando qualquer. Será enviado um ACK caso um comando recebido por algum dos lados tenha sido entendido corretamente, e um NACK caso não tenha sido entendido. Caso o receptor não tenha entendido a mensagem (ou seja, caso NACK), ele enviará NACK ao transmissor e o transmissor desse comando deverá reenviar a mensagem novamente imediatamente.

Exemplo: o PC envia ao arduino o comando STATE, o arduino responde ao PC com a string do comando STATE. O PC verifica que o checksum da mensagem recebida está errado e por isso envia ao arduino o comando NACK. O arduino ao receber o comando NACK refaz e reenvia a string de comando STATE ao PC. O PC verifica que a mensagem veio com o checksum correto, e por isso envia o comando ACK ao arduino. A partir daí a comunicação continua normalmente.

A Tabela1 mostra as possibilidades de envio de comandos para cada plataforma. Um V significa que a referida plataforma pode enviar o comando em questão, um F significa que essa plataforma não pode enviar esse comando, apenas receber.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **COMANDO** | | | | | |
| **PLATAFORMA** | **SET** | **STATE** | **Resposta ao STATE** | **POWER** | **ACK** | **NACK** |
| **PC** | V | V | F | V | V | V |
| **Arduino** | F | F | V | F | V | V |

**Tabela 1 – Possibilidades de comandos por plataforma.**