**CXP – Programa Caixa Preta (06/04/2020)**

Versão 1.0, 14/04/2020

São duas as opções principais: Operação e Teste.

**Operação**: Ao ligar, entra normalmente em operação. Permite escolher um tipo de Operação para a Caixa Preta.

**Teste**: Entra em Teste se ligar a Caixa Preta com a tecla SEL acionada. Permite escolher rotinas para testar os diversos dispositivos da Caixa Preta e fazer novos ensaios.

----------------------- Distribuição da SRAM de 256 KB ---------------------------------

SRAM = 256 KB = 262.144

Mensagem GPS = 128 Bytes

Mensagem MPU = 18 bytes

Configuração = 416 bytes

Qtd msg MPU = 12.720 🡪 12.720 \* 18 = 228.960 Bytes (127,2 segundos)

Qtd msg GPS = 256 🡪 256 \* 128 = 32.768 Bytes (256 segundos)

262.144 – (256+32.768 + 228.960) = 160 bytes sobrando

Mapa da SRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Finalidade** | **Faixa Hexa** | **Bytes** | **Qtd msg** | **Tempo** |
| MPU | 0 0000 🡪 3 7E5F | 228.960 | 12.700 | 127 seg |
| Configuração | 3 7E60 🡪 3 7FFF | 416 | - | - |
| GPS | 3 8000 🡪 3 FFFF | 32.768 | 256 | 256 seg |

#define MPU\_ADR\_INI 0x00000L

#define MPU\_ADR\_FIM 0x37E60L

#define CXP\_ADR\_INI 0x37E60L

#define CXP\_ADR\_FIM 0x38000L

#define GPS\_ADR\_INI 0x38000L

#define GPS\_ADR\_FIM 0x40000L

Funções

|  |  |  |
| --- | --- | --- |
| void | setup | (void) |
| void | loop | (void) |
| byte | sel\_modo | (char \*msg[], byte total) |
|  |  |  |
|  |  |  |

* void **setup** (void)
* void **loop** (void)
* byte **sel\_modo** (char \*msg[], byte total)

Retorna o número da opção selecionada. Recebe um vetor de ponteiros para as mensagens de cada opção e o total de opções. É usada para selecionar o modo de Operação e o tipo de Teste.

**Defs.h**

// TESTE

#define TESTE\_TOT 17 //Modos de teste: 1, 2 , ..., 17

#define TESTE\_0 0 //Não tem

#define TESTE\_1 1 //LEDs

#define TESTE\_2 2 //LCD

#define TESTE\_3 3 //Teclado

#define TESTE\_4 4 //TWI

#define TESTE\_5 5 //Acel e giro

#define TESTE\_6 6 //Magnetometro

#define TESTE\_7 7 //SRAM

#define TESTE\_8 8 //FLAH

#define TESTE\_9 9 //GPS Tudo

#define TESTE\_10 10 //GPS RMC GSA

#define TESTE\_11 11 //GPS U-Center

#define TESTE\_12 12 //MPU-->Matlab

#define TESTE\_13 13 //BlueTooth

#define TESTE\_14 14 //Livre

#define TESTE\_15 15 //Livre

#define TESTE\_16 16 //Livre

#define TESTE\_17 17 //Livre

// OPERA

#define OPERA\_TOT 7 //Modos de teste: 1, 2 , ..., 7

#define OPERA\_0 0 //Não tem

#define OPERA\_1 1 //Livre

#define OPERA\_2 2 //Livre

#define OPERA\_3 3 //Livre

#define OPERA\_4 4 //Livre

#define OPERA\_5 5 //Livre

#define OPERA\_6 6 //Livre

#define OPERA\_7 7 //Livre

**Globs.h**

// TESTE - Mensagens do modo de teste

char \*teste\_msg[]={ "ERRO", //0

"1-LEDs", //1

"2-LCD", //2

"3-Teclado", //3

"4-TWI (I2C)", //4

"5-Acel e giro", //5

"6-Magnetometro", //6

"7-SRAM", //7

"8-FLASH", //8

"9-GPS: Tudo", //9

"10-GPS: RMC GSA", //10

"11-GPS:U-Center", //11

"12-MPU-->MatLab", //12

"13-Blue Tooth", //13

"14-Vazio", //14

"15-Vazio", //15

"16-Vazio", //16

"17-Vazio"}; //17

// OPERA - Mensagens do modo de Operação

char \*opera\_msg[]={ "ERRO", //0

"1-Vazio", //1

"2-Vazio", //2

"3-vazio", //3

"4-Vazio", //4

"5-Vazio", //5

"6-Vazio", //6

"7-Vazio"}; //7

**CXP**

// Selecionar o modo

// Serve selecionar modo de Operação

// Serve selecionar modo de Teste

// A linha 0 é preparada por quem chama

// Usa as linhas 1, 2 e 3

byte sel\_modo(char \*msg[], byte total){

byte prov=total; //provisório = 1, 2, ..., OPERA\_TOT

byte tecla,aux;

while(TRUE){

lcdbx\_apaga\_lin(1);

lcdbx\_apaga\_lin(2);

lcdbx\_apaga\_lin(3);

lcdbx\_str(2,0,"-->");

aux=prov;

lcdbx\_str(1,3,msg[aux++]);

if (aux>total) aux=1;

lcdbx\_str(2,3,msg[aux]);

ser\_str(msg[aux++]); ser\_crlf(1);

if (aux>total) aux=1;

lcdbx\_str(3,3,msg[aux]);

//Esperar tecla

while ( sw\_tira(&tecla) == FALSE);

switch(tecla){

case SW\_SUP: prov--; break;

case SW\_INF: prov++; break;

case SW\_SEL: if (++prov>TESTE\_TOT) prov=1;

return prov;

}

if (prov>total) prov=1;

if (prov==0) prov=total;

}

}