Teste dos dispositivos da Caixa Preta

(04/08/2020)

Opções de teste

|  |  |  |
| --- | --- | --- |
| Nr | Nome | O que faz |
| 1 | Leds | Faz um contador binário com os leds |
| 2 | LCD | Mostra todos os caracteres do LCD |
| 3 | Teclado | Mostra os códigos das teclas |
| 4 | TWI (I2C) |  |
| 5 | Acel e Giro |  |
| 6 | Magnetômetro |  |
| 7 | SRAM |  |
| 8 | FLASH |  |
| 9 | GPS: Tudo |  |
| 10 | GPS: Interpreta |  |
| 11 | GPS: u-Center |  |
| 12 | MPU-->Matlab |  |
| 13 | Blue Tooth |  |
| 14 | BT – Cmds AT |  |
| 15 | Vazio |  |
| 16 | Vazio |  |
| 17 | Vazio |  |
|  |  |  |

// TESTE - Mensagens do modo de teste

char \*teste\_msg[]={ "0-Vai para Opera", //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: Interpreta", //10

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

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

"13-Blue Tooth", //13

"14- BT - Cmds AT", //14

"15-Vazio", //15

"16-Vazio", //16

"17-Vazio"}; //17

A função **byte teste\_sel(void)** permite fazer a seleção entre as opções

**Teste 1 – Leds**

Faz um contador binário com os 4 leds. Usa a função **leds\_cont(char val)**

|  |  |  |  |
| --- | --- | --- | --- |
| Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Laranja | Verde | Amarelo | Vermelho |
| AMO | VD | AM | VM |

**Teste 2 – LCD**

Mostra todos os caracteres do LCD, de 0x00 até 0xFF. Por vez são 64 caracteres, 16 por cada linha. Na coluna da esquerda apresenta em Hexa o código do primeiro caracter. Faz pausa de 1 segundo antes e mudar a tela.

**Teste 3 – Teclado**

Na porta serial é possível ver os transientes que surgem quando uma tecla é acionada ou solta.

A variável **byte sw\_val** é atualizada pelo Timer 1 e contém a média das duas últimas leituras (8 bits) feitas com o ADC.

**Teste 12 – MPU 🡪 Matlab**

Envia parâmetros de configuração e depois os dados, na taxa de 100 Hz.

Data = 010420

Hora = 123456.78

Detalhamento dos dados enviados por essa opção. Todos os dados são strings ASCII delimitados por espaços em branco ou fim de linha (0xD 0xA).

|  |  |
| --- | --- |
| **Linha** | **Descrição** |
|  | **------- dados do MPU ------** |
| **#[m** | Delimitador de início do campo de dados do MPU. |
| ddmmyy | Data da batida (do acionamento da tecla SEL) (string). |
| hhmmss | Hora da batida (do acionamento da tecla SEL) (string). |
| temp | Leitura do registrador de temperatura na hora do acidente. |
| st\_op | Resultado do Self Test ao ligar (acelerômetro e giroscópio) |
| sth\_op | Resultado do Self Test ao ligar (magnetômetro) |
| adr\_mpu | Ponteiro de gravação dos dados do MPU na hora da batida. |
| adr\_gps | Ponteiro de gravação dos dados do GPS na hora da batida. |
| cf\_ok | 1.Calibração de Fábrica foi realizada? (sim = 21.331= ‘SS’ e Não = 20.046 = ‘NN’)  2.Passou no Self Teste da Calibração de Fábrica? (sim = 21.331= ‘SS’ e Não = 20.046 = ‘NN’)  3.Escala do Acelerômetro na Calibração de fábrica (1, 2, 4, 8 g)  4.Escala do Giroscópio na Calibração de Fábrica (250, 500, ..., 2000 graus/seg)  5.Leitura do registrador de temperatura durante a Calibração de Fábrica  6.Offset de ax (Calibração de Fábrica)  7.Offset de ay (Calibração de Fábrica)  8.Offset de az (Calibração de Fábrica)  9.Offset de gx (Calibração de Fábrica)  10.Offset de gy (Calibração de Fábrica)  11.Offset de gz (Calibração de Fábrica) |
| st\_cf |
| aesc\_cf |
| gesc\_cf |
| tp\_cf |
| ax\_off |
| ay\_off |
| az\_off |
| gx\_off |
| gy\_off |
| gz\_off |
| cfh\_ok | 1.Feita Calibração de Fábrica do Magnet? (sim = 21.331= ‘SS’ e Não = 20.046 = ‘NN’)  2.Passou Self Teste do Magnetômetro na CF? (sim = 21.331= ‘SS’ e Não = 20.046 = ‘NN’)  3.Ajuste do eixo hx  4.Ajuste do eixo hy  5.Ajuste do eixo hz  6.Offset de hx (Calibração de Fábrica)  7.Offset de hy (Calibração de Fábrica)  8.Offset de hz (Calibração de Fábrica)  9.Ajuste de escala de hx (Calibração de Fábrica)  10.Ajuste de escala de hy (Calibração de Fábrica)  11.Ajuste de escala de hz (Calibração de Fábrica) |
| sth\_cf |
| hx\_ASA |
| hy\_ASA |
| hz\_ASA |
| hx\_off |
| hy\_off |
| hz\_off |
| hx\_esc |
| hy\_esc |
| hz\_esc |
| st\_op | 1. Passou Self Teste da Operação? (sim = 21.331= ‘SS’ e Não = 20.046 = ‘NN’)  2. Magnet passou Self Teste da Operação? (sim = 21.331= ‘SS’ e Não = 20.046 = ‘NN’)  3.Escala do Acelerômetro na Operação (1, 2, 4, 8 g)  4.Escala do Giroscópio na Operação (250, 500, ..., 2000 graus/seg)  5. Frequência de amostragem em Hz  6. BW usada na operação  7. Reserva  8. Reserva  9. Reserva  10. Reserva  11. Reserva |
| sth\_op |
| aesc\_op |
| gesc\_op |
| famost |
| bw |
| r1 |
| r2 |
| r3 |
| r4 |
| r5 |
| qtd | Quantidade de linhas com as informações do MPU (ax, ay, ..., hz). |
| cont adr  ax ay ax  gx gy gz  hx hy hz | Linha com uma leitura do MPU. Todos esses dados estão em uma mesma linha, separados por espaços em branco.  cont é um contador sequencial.  adr é o endereço em decimal da posição onde este dado do MPU estava na SRAM. |
| **m]#** | Delimitador de fim do campo de dados do MPU |

#[m

040820

181525

1930

20046

20046

43182

232448

+21331 +20046 +2 +250 +2111 +544 +232 +17905 +124 +112 +62

+21331 +181 +181 +183 +171 -100 +2005 -680 +1940 +3255 +3230

+20046 +20046 +2 +250 +100 +5 +1 +2 +3 +4 +5

12720

... ax ay az gx gy gz hx hy hz

ax\_off, ay\_off, az\_off, gx\_off, gy\_off, gz\_off 🡪 Calibração de Fábrica

tp\_cf 🡪 Temperatura por ocasião da calibração de fábrica

a\_esc 🡪 escala do acelerômetro +/-2, +/-4, +/-8, +/-16,

g\_esc 🡪 escala do giroscópio +/-250, +/-500, +/-1000, +/-2000,

famostr 🡪 Frequência de amostragem, em princípio 100 Hz

bw 🡪 Banda do filtro do MPU

hx\_off hy\_off hz\_off 🡪 off set

hx\_esc hy\_esc hz\_esc 🡪 escala

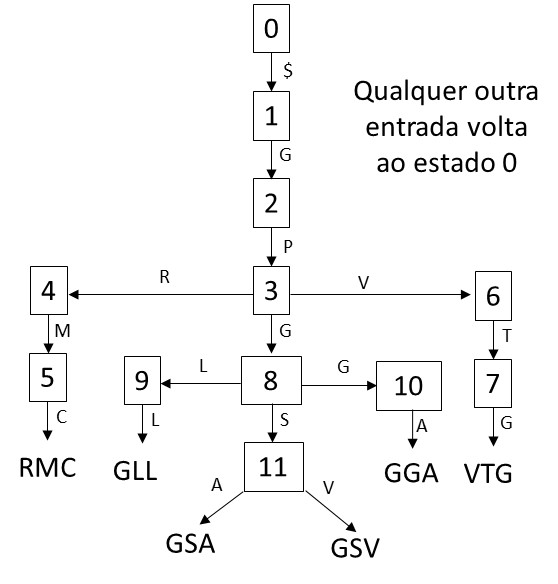
hx\_asa hy\_asa hz\_asa 🡪 ASA

**ASAX, ASAY, ASAZ:** Sensitivity Adjustment values (ASA tem apenas 1 byte e parece não ter sinal). Exemplo para hx, idem para hy e hz.

**CALIBRAÇÃO**: Para ter um pouco de precisão, os valores de offset e ajuste estão multiplicados por 10. Então, primeiro é preciso dividi-los por 10 antes de usá-los.

**Teste 9 – GPS Tudo**

Diagrama de estados para identificar as mensagens.



**Teste 10 – Filtrar informações de RMC e GSA**

Apresenta no LCD:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| D | D | / | M | M | / | A | A | b | H | H | : | M | M | : | S | S | b | N | N |
| b | La | La | b | La | La | . | La | La | La | La | La | NS |  | P | P. | . | P | P | p |
| Lo | Lo | Lo | b | Lo | Lo | . | Lo | Lo | Lo | Lo | Lo | EW |  | A | A | . | A | A | A |
| V | V | V | . | V | V | V | k | p | h |  | C | C | C | . | C | C | C |  | ST |

DD/MM/AA 🡪 dia, mês e ano HH:MM:SS 🡪 hora, minuto, segundo NN🡪 quantidade de satélites

LaLa LaLa.LaLaLaLaLaLa (DDMM.MMMMM) NS 🡪 Latitude CCC.CCC 🡪 Curso

LoLoLo LoLo.LoLoLoLoLo (DDDMM.MMMMM) EW 🡪 Longitude ST🡪Status

VVV.VVV 🡪 velocidade em kph PP.PPP 🡪 PDOP = espalhamento 3D AA.AAA 🡪 Altitude

No monitor serial apresenta:

gps\_dados: Status=A Hora=011507.00 Data=140420 Lat=1548.62918 N/S=S Long=04748.65826 E/W=W Vel Nos=0.645 Curso= PDOP=4.79 HDOP=2.45 VDOP=4.12 Vel\_kph=1.195 Vel\_uni=K Qtd\_Sat=05 Alt=1060.9 Alt\_uni=M

vetor **gps\_dados**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RMC | RMC | RMC | RMC | RMC | RMC | RMC |
| 0 | 2 | 13 | 20 | 31 | 33 | 45 |
| Status | Hora | Data | Latitude | N/S | Longitude | E/W |
| A0 | hhmmss.sss0 | ddmmyy0 | ddmm.mmmmm0 | N0 | dddmm.mmmmm0 | E0 |
| A | 083559.00 | 091202 | 4717.11437 | N | 00833.91522 | E |
| 1+1 | 10+1 | 6+1 |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RMC | RMC | GSA | GSA | GSA | VTG | VTG |
| 47 | 55 | 63 | 69 | 75 | 81 | 88 |
| Velocidade Nós | Curso | PDOP | HDOP | VDOP | Speed km/h | Unidade |
| ddd.ddd0 | ddd.ddd0 | dd.dd0 | dd.dd0 | dd.dd0 | xxx.xx0 | K0 |
| 0.004 | 77.52 | 99.99 | 99.99 | 99.99 | ?125.12 | K |
| 1+1 |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GGA | GGA | GGA | GGA |  |  |  |
| 90 | 92 | 95 | 101 | 104 |  |  |
| Fix | Qtd Sat | Altitude | Unidade |  |  |  |
| d0 | dd0 | dddd.d0 | m0 |  |  |  |
| 1 | 4 | 627.4 | m |  |  |  |
| 1+1 | 2+1 |  | 1+1 |  |  |  |

**Teste 11 – GPS 🡪 u-Center**

**Teste 12 – MPU 🡪 Matlab**

Aqui existe um problema: quando o Matlab abre a porta serial (COMi), o Arduino Resseta. Isso acontece porque ao abrir a porta COMi, ele desativa e ativa novamente o DTR. Isto significa Reset para o Arduino. O Matlab, espera receber dados e, no entanto, chegam mensagens das opções de teste.

A solução foi colocar pausas e exigir uma sequência para sincronizar Arduino e Matlab.

|  |  |
| --- | --- |
| Matlab | Arduino |
| Abre a porta COMi | RESET |
| Mensagem: Prepare Arduino  Pausa (10) | Tempo para o Arduino ressetar e permitir ao usuário posicionar para a opção 12 |
| flushinput(sid) = apaga o passado serial | - |
| Mensagem: Pronto para receber dados. | (SEL) Seleciona a opção 12 |
| Pausa (4) | Inicia a transmissão |
| fprintf(1,'Recebendo ...\n');  flushinput(sid); |  |

; Exemplo Teraterm - força reset do Arduino

; enable dtr

setdtr 1

; clear rts

;setrts 0

; now clear dtr, so that the system will reset and see rts clear

setdtr 0

**Sugestão 1)**

No futuro usar duas posições da memória SRAM para indicar que o Arduino foi ressetado pelo Matlab. Por exemplo, quando entrar no modo que envia para o Matlab, gravar 0x55 e 0xAA nas posições 0 e 1 da SRAM. Ao acorda após o reset, ele verifica essas duas posições e se forma0x55 e 0xAA, entra diretamente no modo que envia para o Matlab. Dispensa ação do usuário para selecionar novamente e evita o envio de texto pela porta serial, o que confunde o Matlab.

**Sugestão 2)**

O Matlab abre a porta COMi e fica esperando uma sequência especial antes de iniciar a recepção das leituras do MPU. O Arduino após entrar novamente na opção correta, transmite a sequência especial, por exemplo 0x55 e 0xAA e em seguida inicia o envio dos dados.

**Teste 13 – Bluetooth**

É preciso parear o computador com o HC-05. Senha “1234”.

Porta serial configurada para 115.200 bps.

Repetidamente, transmite as letras de A até Z, CR e LF.

Ecoa no terminal serial e no LCD.

Usar um terminal serial (TeraTerm) ou uma outra sessão do Arduino para acompanhar o que chega pelo BueTooth.

**Teste 14 – BT – Cmds AT**

Opção para permitir enviar comandos AT para o HC-05.

Não precisa parear o computador com o HC-05.

PC <==> Arduino (COM0) opera em 115.200 bps.

Arduino (COM2) <==> HC-05 opera em 38.400 bps.

Cada comando precisa terminar com Carriage Return (CR=0xD) e Line Feed (LF=0xA)

Por isso, no terminal serial do Arduino, habilitar <Ambos, NL e CR>.

Usar este modo para configurar HC-05:

AT+UART=115200,0,0

AT+NAME=Caixa Preta

Como colocar HC-05 no modo comando:

1. Remova o jumper J1
2. Acione o pequeno botão do HC-05
3. Recoloque o (J1) jumper de alimentação
4. Solte o pequeno botão do HC-05
5. Led passa piscar lentamente
6. HC-05 opera em 38.400 bps

Exemplos:

|  |  |
| --- | --- |
| Comando | Resposta |
| AT | OK |
| AT+UART? | +UART:115200,0,0  OK |
| AT+UART=115200,0,0  (erro: no doc pdf tem um vírgula a mais) | OK |
| AT+PSWD? | +PIN:"1234"  OK |
| AT+VERSION? | VERSION:3.0-20170601  OK |
| AT+ADDR?  (endereço ?Bluetooth) | +ADDR:98D3:31:F5B0E1  OK |
| AT+NAME? | +NAME:HC-05 |
| AT+NAME=Caixa Preta | OK |
| AT+NAME? | +NAME:Caixa Preta |