**MINERVA ROCKETS**

**GUIA PDA v0.3**

**Por Henrique Bruno**

**1. Sobre**

O projeto PDA foi um projeto de recepção e de apresentação gráfica de dados do foguete, enviados por telemetria via Rádio Frequência.

Optamos por usar a plataforma Raspberry Pi 3 (abreviado neste documento como somente “RPi”), devido à diversos fatores, sendo alguns deles: disponibilidade em mãos do dispositivo, facilidade de reposição em caso de problemas, domínio prévio da plataforma, processador razoavelmente rápido (4× ARM Cortex-A53, 1.2GHz), e a intenção de no futuro utilizar um pequeno monitor HDMI para a apresentação dos dados, visto que será mais veloz mudar apenas parcelas do código para adequar o monitor, do que trocar de plataforma e escrever um sistema novo.

O PDA apresenta dois displays gráficos monocromáticos de 128x64 pixels, sendo as funções principais a plotagem de diversos gráficos, um sistema GPS que compara a posição deste monitoramento terrestre com a posição do foguete, assim mostrando sua localização e distância em metros e um menu para controle do sistema. Apresenta também, um display de caracteres 20x04, que mostra status do sistema de telemetria e o apogeu do foguete. Essas informações podem ser facilmente trocadas pelo programador, inclusive pode-se ainda acrescentar outros displays gráficos ou de caracteres, sem grandes dificuldades, podendo-se assim alterar a quantidade de informações sendo exibidas. O PDA ainda utiliza um magnetômetro, que funciona em conjunto com o sistema de GPS, onde aponta para onde o dispositivo está virado.

A linguagem de programação utilizada foi o Python 2.7.9. O motivo principal da escolha desta linguagem é que Python é a linguagem com mais “bibliotecas” (chamado no Python como *package*) disponíveis para o RPi, facilitando e agilizando a construção do código, mesmo que seu desempenho não seja tão bom quanto usar por exemplo a linguagem C.

**2. Dados técnicos**

|  |  |
| --- | --- |
| Plataforma | **Raspberry Pi 3** |
| GPS | **NEO6MV2** |
| Magnetômetro (utilizado um IMU) | **GY-801** |
| Módulo Rádio Frequência | **LoRa RFM95W** |

Os displays sendo genéricos, não possuem um modelo específico.

Alimentação ideal do RPi: 5V @3A

Tensão de funcionamento do RPi: 3.3V (não aplicar 5V nas GPIO)

**3. Utilização**

1. Teclas de interação com os displays gráficos:

|  |  |
| --- | --- |
| **Esc** ou **i** | Abre/fecha o menu |
| Seta para cima, seta para baixo | Destaca opção no menu |
| **Enter** | Seleciona opção destacada no menu |
| **Ctrl** | Alterna o display gráfico ativo |

1. É possível efetuar o desligamento do sistema de duas maneiras:
   1. Abrindo o menu no display gráfico, selecionando “Change mode”, depois “System menu”, e por fim, a opção “Sys shutdown”, e esperar a contagem decrescente para o desligamento, onde é possível durante esta abortar o desligamento com o pressionamento de qualquer tecla de Abrir/fechar menu ou de seleção.
   2. Segurar durante 3 segundos o botão físico presente no PDA.

Nos dois métodos, deverá aparecer um texto no display gráfico explicando que deve-se esperar cerca de 10 segundos até **que possa ser feita a** **retirada segura do cabo de energia** do RPi.

Quando o comando de desligamento for executado, o LED verde do RPi irá piscar 6 vezes rapidamente, e quando este parar de piscar, significa que o sistema está desligado e pode ser feita a remoção do cabo de energia.

1. Este RPi roda o sistema operacional Raspbian, que por sua vez é baseado em Debian (distribuição Linux), logo é possível a execução de comandos sem a presença de um monitor, somente com um teclado.
2. Ctrl + Alt + t = Abre o terminal do Linux
3. Digitar, sem aspas: “sudo killall python” = Encerra todos os processos python em execução (inclui os processos do PDA)

NOTA: NÃO use esse comando ou quaisquer outros comandos que gerem sinais SIGINT, SIGTERM etc, em situações onde há a real leitura do RF e a respectiva gravação no log, pois há grandes chances de corrompimento do arquivo de log. Foi decidido que não seria programado os tratamentos destes sinais, visto que é fácil evitar o possível erro, com o correto desligamento do sistema (3.2)).

1. Ctrl-C = Certifica/encerra qualquer processo sendo executado em primeiro plano. (NOTA: o programa do PDA, executado automaticamente no boot, é executado em segundo plano)
2. Digitar, sem aspas: “sudo shutdown -h now” = Desliga o RPi, e possibilita a remoção segura do cabo de energia.

4) Informações gerais de utilização:

a) O diretório do programa principal é: “/home/pi/pda”.

b) O caminho do programa principal é: “/home/pi/pda/pdaMain.py”

c) Este Raspberry Pi conta com uma configuração de auto execução ao iniciar, definido em:

“/etc/rc.local”

Sendo o comando a ser executado ao iniciar

“sudo python /home/pi/pda/autorun.py &”

d) O programa autorun, executado na inicialização do RPi, além de inicializar o programa principal, ainda possibilita que o RPi seja desligado pelo pressionamento do botão por 3 segundos, com uma espera de 10 segundos até que o comando de shutdown seja executado.

NOTA: O PDA conta com dois códigos para a utilização deste botão, um no programa principal e o outro no autorun. Assim foi feito, para que idealmente o sistema seja desligado a partir do programa principal (que irá parar todas as funções, como a de escrita de log, e desligará corretamente sem chances de corrompimento deste arquivo) , e o autorun seja o código de backup, caso ocorra a finalização indevida do programa principal, o RPi ainda possa ser desligado por este botão.

Definimos também o comando “autorun” para executar as seguintes ordens:

sudo killall -q python; sudo python /home/pi/pda/autorun.py

Os logs de dados da sessão do PDA são salvos no diretório “log”, no diretório principal do programa principal.

**3.1 Bugs conhecidos (LEIA)**

1) Caso seja conectado um teclado, este não poderá ser desconectado da conexão USB até o desligamento do sistema. Entretanto, no caso de teclados sem fio, este poderá ser desligado normalmente, porém sem que o receptor dos dados seja desconectado do USB.

2) Ocasionalmente podem aparecer artefatos gráficos (aparecimento de pixels não esperados / erros gráficos) nos displays.

1. **Conexões**

**1) Conexões dos displays gráficos**

Por mais que os displays gráficos estejam operando em 5V e o RPi funcione à 3.3V, não existem problemas em ligar diretamente os pinos lógicos de saída do RPi diretamente nos displays.

Então, os pinos RST, R/W e E não precisam ser convertidos para 5V.

**a) Display gráfico geral**

|  |  |
| --- | --- |
| **Display 128x64** | **Raspberry** **Pi 3** |
| GND | GND |
| VCC | 5V |
| RS (modo paralelo ou serial, queremos o serial, HIGH) | 3.3V ou 5V |
| PSB | GND |
| RST | GPIO 22 (pin 15) |
| BLA (Luz de fundo) | 5V |
| BLK | GND |

**b) Display gráfico 1**

|  |  |
| --- | --- |
| **Display0 128x64** | **Raspberry** **Pi 3** |
| R/W (Data serial) | GPIO 17 (pin 11) |
| E (Enable/Clock) | GPIO 27 (pin 13) |

**c) Display gráfico 2**

|  |  |
| --- | --- |
| **Display0 128x64** | **Raspberry** **Pi 3** |
| R/W (Data serial) | GPIO 20 (pin 38) |
| E (Enable/Clock) | GPIO 21 (pin 40) |

**2) Conexão do display de caracteres**

**ATENÇÃO:**

É possível utilizar no VCC 5V, para que o brilho seja mais forte. Entretanto, a tensão de operação das GPIO é de 3.3V, e que neste display pode ocorrer uma corrente inversa, o que danificaria o RPi.

Então, caso seja optado em usar 5V, deve ser utilizado um conversor de nível lógico 3.3V para 5V (presente no PDA) nos pinos SDA e SCL, para a utilização segura do mesmo.

Endereço I2C = 0x3D

|  |  |
| --- | --- |
| **Display de caracteres** | **Raspberry** **Pi 3** |
| GND | GND |
| VCC | 5V ou 3.3V (leia a atenção) |
| SDA | SDA1 (pin 3) |
| SCL | SCL1 (pin 5) |

**3) Conexão do GPS**

|  |  |
| --- | --- |
| **GPS** | **Raspberry** **Pi 3** |
| VCC | 3.3V |
| RX | TXD0 (pin 8) |
| TX | RXD0 (pin 10) |
| GND | GND |

**4) Conexão do IMU:**

Endereço I2C = Diversos, porém não é necessário atentar-se à isso.

|  |  |
| --- | --- |
| **IMU** | **Raspberry** **Pi 3** |
| VCC\_3.3V | 3.3V |
| GND | GND |
| SCL | SCL1 (pin 5) |
| SDA | SDA1 (pin 3) |

**5) Conexão do LoRa RFM95W:**

|  |  |
| --- | --- |
| **LoRa** | **Raspberry** **Pi 3** |
| VCC | 3.3V |
| GND | GND |
| RST | GPIO 24 (pin 18) |
| CS (Chip select) | GPIO 8 (CE0, pin 24) |
| G0 (Interrupt) | GPIO 25 (pin 22) |
| MISO | GPIO 9 (MISO, pin 21) |
| MOSI | GPIO 10 (MOSI, pin 19) |
| SCLK | GPIO 11 (SCLK, pin 23) |
| EN | O pino de enable não é conectado. |