

Como criar um VideoGame Portátil - Um guia completo para iniciantes de Cubieboard 2

Eu tenho em minhas mãos um cubieboard, e todo o potencial que ele tem. Você pode fazer o seu próprio Video-Game sistema portátil com essa cubieboard.



Caramba!!! Escreva um programa, ligue-o, e BOOM coisas incríveis acontecem!

Quer dizer, nem sempre.

Eu tive um pouco de dificuldade para programar o meu cubieboard, então aqui está um guia rápido sobre a programação cubieboard

Primeiro, o que é um cubieboard? Eu nunca tinha ouvido falar dele antes, eu poderia ter usado uma breve explicação. Basicamente é o computador que impulsiona tabelas e smartphones. É um pequeno computador programável com pinos de hardware que você pode conectar a qualquer coisa. Talvez você tenha ouvido falar de seu primo menor, o Raspberry Pi. Há um livro no Google sobre o tema, bem escrito e completo para iniciantes .

Então, um cubieboard é um computador, certo? Então eu fiz o que você faz com qualquer computador novo: eu ansiosamente olhei para a fonte de alimentação e conectei-o. Ou tentei. Deixe-me dizer-lhe agora, ele não vem com um cabo de alimentação e não há um interruptor. Ele liga quando você conecta uma fonte de alimentação. Felizmente meu laptop tem fonte de alimentação. Mas, infelizmente, meu cubieboard não veio com qualquer sistema operacional.

É comum vir com o Android pré-instalado. O meu não veio, não sei porque motivo. Não importa, porque você é capaz de instalar um sistema operacional!

Instalar um sistema operacional ?? Não entre em pânico! Não foi tão difícil.

Antigamente, você tinha que carregar um CD com um sistema operacional, entrar nas configurações do BIOS para carregar o CD no boot, redefinir o computador e passar por um processo de várias horas de duração. Um cubieboard é mais simples e mais rápido.

Primeiro você precisa de um cartão "MicroSD" e uma maneira de conectar o cartão MicroSD ao seu computador normal. Eu não tinha um. Eu tive que correr para Walmart. Conecte o cartão MicroSD ao seu computador e siga estas instruções para copiar o Cubian para ele. Cubian é um sistema operacional especificamente para cubieboards. É uma versão modificada do Linux. Se você planeja fazer qualquer programação em um cubieboard você tem que utilizar o Cubian. Não siga outros tutoriais dizendo que você deve instalar Debian, ou Ubuntu, ou BootBerry. Eu tentei esses. O Debian produziu um flicker tortuoso, e o Ubuntu, enquanto ele parecia muito bonito e fácil de usar, deu problema depois de uma hora. BootBerry, enquanto escrito para Raspberry Pi, funcionou bem, mas não fornece uma opção para instalar Cubian. O cubian tem tudo pré-configurado.

Você não precisará lidar com arquivos FEX ou kernels (saiba mais sobre isso mais tarde).

Este guia presume que você vai usar os pinos no cubieboard, para conectá-los a botões ou sensores ou o que quer que seja . Há algumas coisas que você precisa saber sobre os pinos no cubieboard, coisas que me levou um tempo para descobrir.

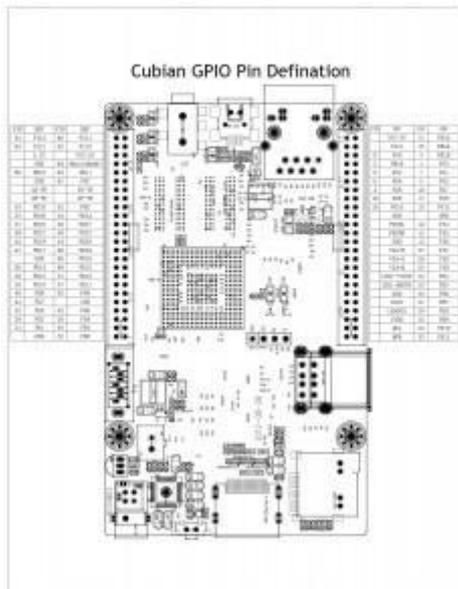
Primeiro, os pinos NÃO são GPIOs. Os pinos são os pedaços físicos do metal saindo da placa. GPIOs são coisas abstratas que devem ser criadas. Eles podem ser pinos, mas para o cubieboard você tem que atribuir um pino para um GPIO.

Em segundo lugar, existem 96 pinos, mas você só pode usar 60-alguns deles. Por quê? Não tenho certeza. Alguns pinos carregam a tensão pura, ou conectam na base, que é porque não podem ser usados como entradas / saídas, mas aquele explica somente aproximadamente 5 pinos. Os outros 25+ não tenho idéia .

Em terceiro lugar, pinos e GPIOs têm um sistema de numeração diferente. Pin 1 não é o mesmo que GPIO 1. Na verdade, Pin 1 no cubieboard é uma conexão direta com a fonte de alimentação, você não pode usá-lo como um GPIO.

Em quarto lugar, você deve atribuir pinos a GPIOs, por algo chamado de arquivo "Fex", às vezes referido como "script.fex" ou "script.bin". Isso é feito automaticamente se você instalou o Cubian.

Em quinto lugar, os pinos (e por proxy, o GPIOs) têm nomes estranhos aleatórios. Os pinos, enquanto numerados, têm nomes como "CSI1", "DEF", "VCC-1", etc. Pinos utilizáveis, pinos que você pode atribuir a GPIOs, têm nomes como "PG3", "PD5", "PE9" Todos começam com P).



Sexto, você precisa saber todos os 5 passos anteriores. Desculpe, é verdade. Você precisa saber os nomes dos pinos, os números de pinos e qual pino corresponde a qual GPIO. Felizmente, há um gráfico útil encontrado aqui.

Cubieboards são burros. Eles não podem detectar seus próprios pinos. Eles não sabem quais pinos são entradas ou saídas. Eles não sabem nada. Você precisa instalar um sistema operacional apenas para obter um cubieboard para fazer qualquer coisa, e os únicos sistemas operacionais livres (Linux, Android) não são destinados a trabalhar com pinos físicos. Se você não usasse o Cubian, você teria que instalar um software especial (chamado de Kernel) para obter o Sistema Operacional para acessar os pinos, então você precisaria instalar um software especial para acessar os arquivos que acessam os pinos, então você precisa Editar os arquivos .

Tudo isso era informação de background para um grande segredo: pinos em um cubieboard são acessados via arquivos. Edite o arquivo no sistema operacional e a tensão no cubieboard muda. Ou, se for uma entrada, mude a tensão e o arquivo muda. Levei um dia para perceber isso, então eu vou repeti-lo. Todos esses tutoriais sobre programação GPIOs e trechos de código são sobre uma coisa.

E uma coisa só - alguém inventou uma maneira de acessar cada GPIO em um cubieboard, modificando um arquivo regular.

Experimente você mesmo, agora mesmo. Supondo que você tenha cubian instalado, digite:

```
echo timer > /sys/class/leds/blue:ph21:led2/trigger
```

E seu cubieboard deve começar a piscar (a luz azul). Se você não tiver cubian você terá que instalar um pacote chamado sunxi-leds.

O que esta acontecendo aqui? Bem, se você conhece o Linux, sabe que "echo" apenas produz algum texto - "timer" é uma palavra de código especial – e o símbolo ">" significa apenas "write (ou" pipe ") para este arquivo. O arquivo é chamado de "trigger" e está no diretório / sys / class / leds / blue: ph21: led2. Sim, esse é um nome de diretório válido no Linux. Então estamos colocando a palavra "timer" no arquivo chamado "trigger" e o cubieboard automaticamente começa a piscar! É brilhante, e funciona para todos os GPIO no cubieboard.

Infelizmente, esses arquivos especiais não existem automaticamente. Eles fazem para o exemplo acima, a luz azul piscando, e a luz verde, mas você deve criá-los para cada pino ou GPIO. Então aqui estão os passos para habilitar um GPIO hipotético # 1:

No prompt de comando do linux, digite:

```
echo 1> / sys / class / gpio / export
```

(Note que o echo 1 corresponde ao número do GPIO que você deseja usar.) Se você quiser criar, digamos, GPIO # 21, digite 'echo 21').

Este é o comando que cria os arquivos mágicos que se conectam aos pinos. Se você digitar 'ls' no prompt de comando, você verá um novo diretório criado. Infelizmente, o novo diretório é nomeado após o GPIO que você criou e o nome do pino físico. Assim, o novo diretório será chamado algo como "gpio1_pg3" onde a primeira parte é o nome do GPIO (que você criou) e a segunda parte é o nome do pino (que você tem que memorizar a partir do gráfico). É estúpido, mas é o que você pode fazer.

Agora, decida se este GPIO vai ser uma entrada ou saída. Ou seja, você quer que a tensão vá para este pino mude com base no que o programa de computador está dizendo o que fazer (saída), ou é o pino que vai receber a tensão de alguma fonte externa e um programa de computador para detectá-lo (entrada) ?

Se for de entrada, digite

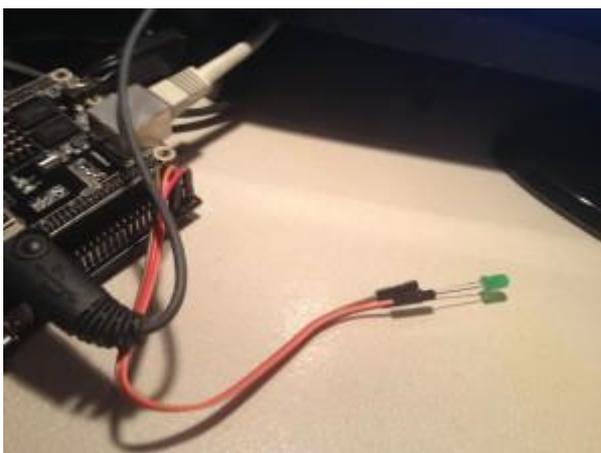
```
echo "in"> / sys / class / newdirectory / direction
```

Ou para saída,

```
echo "out"> / sys / class / newdirectory / direction
```

Agora, você pode definir o valor do pino / GPIO diretamente (se ele é saída) ou detectá-lo usando o comando cat (se ele for entrada).

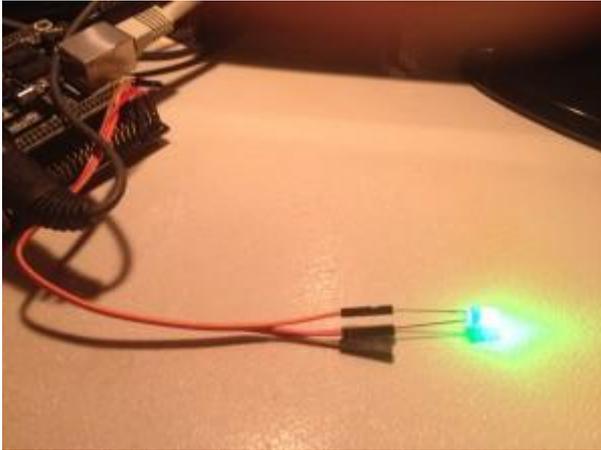
Para testar isso, você precisará de um LED / diodo e um cabo de jumper pequeno. Conecte o pino # 1, que é uma conexão direta com a fonte de alimentação, com o seu agora pino GPIO # 1 (chamado 'pg3', quarto a partir da linha superior direita, veja diagrama.) Conecte a extremidade curta do LED / Diodo para pino 1 (a tensão contínua) e a extremidade longa para pino 'pg3' (novamente, 4 ° a partir do canto superior direito)



Em seguida, no prompt de comando, digite

```
cd /sys/class/gpio/gpio1_pg3  
echo 1> value
```

E o LED / diodo deve acender. (Imagem)



Incrível, certo? Até agora você deve perceber que

```
echo 0 > value
```

Irá desligá-lo.

Se você fez isso até aqui, e entendeu tudo até este ponto, tudo que você precisa fazer é entender como fazer tudo programaticamente. Eu sou treinado na linguagem C, então eu vou fazer meus exemplos em C. A maioria dos exemplos lá fora são em Python. Use qualquer linguagem de programação que você preferir.

Você precisará saber como criar e compilar um programa em C. Então instale o compilador C, então usando o seu editor de texto preferido (eu prefiro o nano simples) crie seu programa C (chamado 'test.c'), então Compile com o comando `gcc test.c`, em seguida, execute-o digitando `./a.out`. Aqui está o meu programa completo para ligar o LED:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <stdlib.h>
```

```
int main(){
```

```
FILE * fp;
```

```
if((fp = fopen("/sys/class/gpio/gpio1_pg3/value", "w")) == NULL){
```

```

printf("File not found");

exit(1);

}

fprintf(fp, "1");

fclose(fp);

return 0;

}

int main () {

FILE * fp;

if (fp = fopen ("/ sys / class / gpio / gpio1_pg3 / value", "w")) == NULL) {

printf ("Arquivo não encontrado");

Saída (1);

}

fprintf (fp, "1");

fclose (fp);

Return 0;

}

```

A chave é usar o comando fprintf ().

Para detectar se uma tensão de pino foi alterada (ou seja, o pino é uma entrada) você faz um processo semelhante - Abra o arquivo e verifique se o valor é 0 ou 1. Você pode fazer isso no linux indo para o diretório / arquivo e Digitando no valor do Cat (veja o vídeo). 1 é o padrão, e quando você ligar o pino para base (que é algo como o pino # 20), o valor mudará para 0. (note isso é diferente para a saída - saída você anexar o pino para VCC (pino # 1) , Não terra).

Em C, use o comando fgetc ():

```
Intch;
```

```
Fd = open (filename, O_RDONLY);
```

```
Ch = fgetc (fd);
```

```
If (ch! = '0') {
```

```
  //stuff
```

```
} else {
```

```
  //stuff
```

```
}
```

```
close (fd);
```

Espero que ajude iniciantes com cubieboard. Boa sorte!