

Manual de Usuário

Projeto: FIREFLY

Colaborador: Diogo Batista dos Santos

Histórico de Revisões

Data	Versão	Descrição	Autor
01/12/2018	1.0	Criação do documento	Diogo B. dos Santos

Sumário

1. INFORMAÇÕES BÁSICAS	4
2. HARDWARE	5
2.1. NVIDIA Jetson TX1	5
2.2. DLP4500	7
2.3. Bridge	8
3. SOFTWARE	12
4. ARQUIVOS E LINKS	14

1. INFORMAÇÕES BÁSICAS

Bem-vindo ao manual de usuário do projetor Firefly. Esse documento ajuda o usuário com a instalação e a configuração inicial dos módulos individuais e do conjunto como um todo.

O Firefly (figura 1) é composto de três placas principais: o módulo NVIDIA Jetson TX1, o projetor DLP4500 e a placa de comunicação Bridge. Esse documento aborda a configuração de cada placa, bem como a conexão entre elas, assim como o software de demonstração utilizado na TX1.

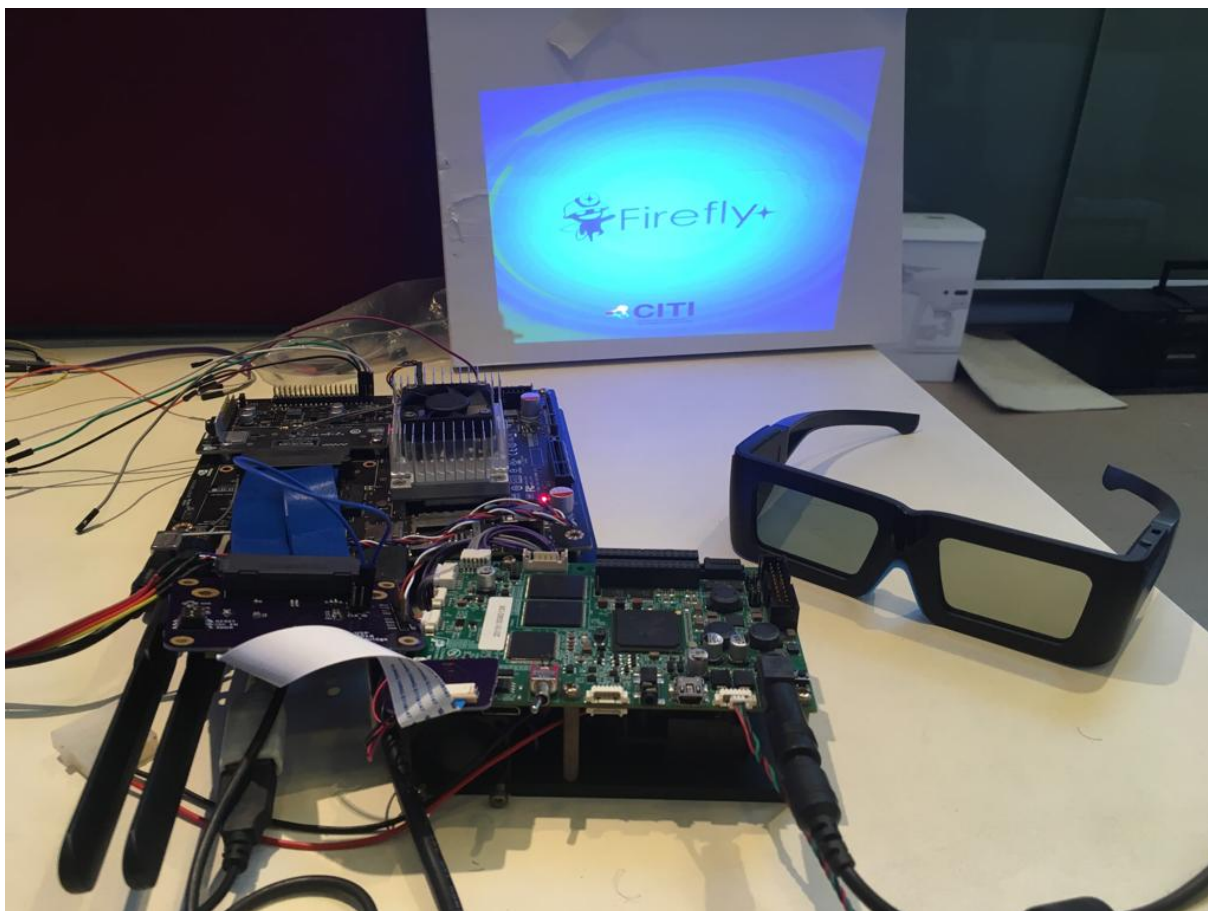


Figura 1: O Firefly.

2. HARDWARE

2.1. NVIDIA Jetson TX1

O módulo TX1 é um Computer-on-module, um sistema embarcado híbrido entre microcontrolador e computador completo, contendo microprocessador, memórias, entradas e saídas e outros periféricos. Ideal para usos em aplicações pequenas ou especializadas, pois apresenta um consumo de energia menor do que computadores completos e uma capacidade computacional maior que os microcontroladores.

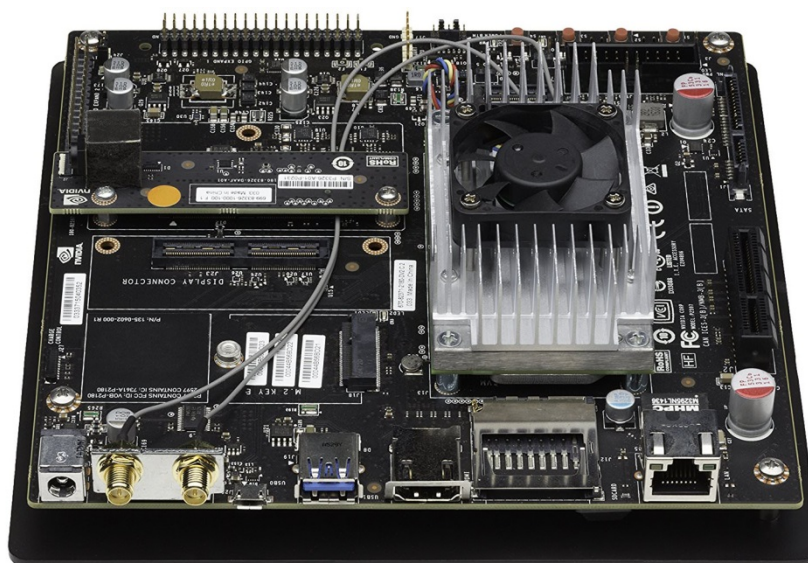
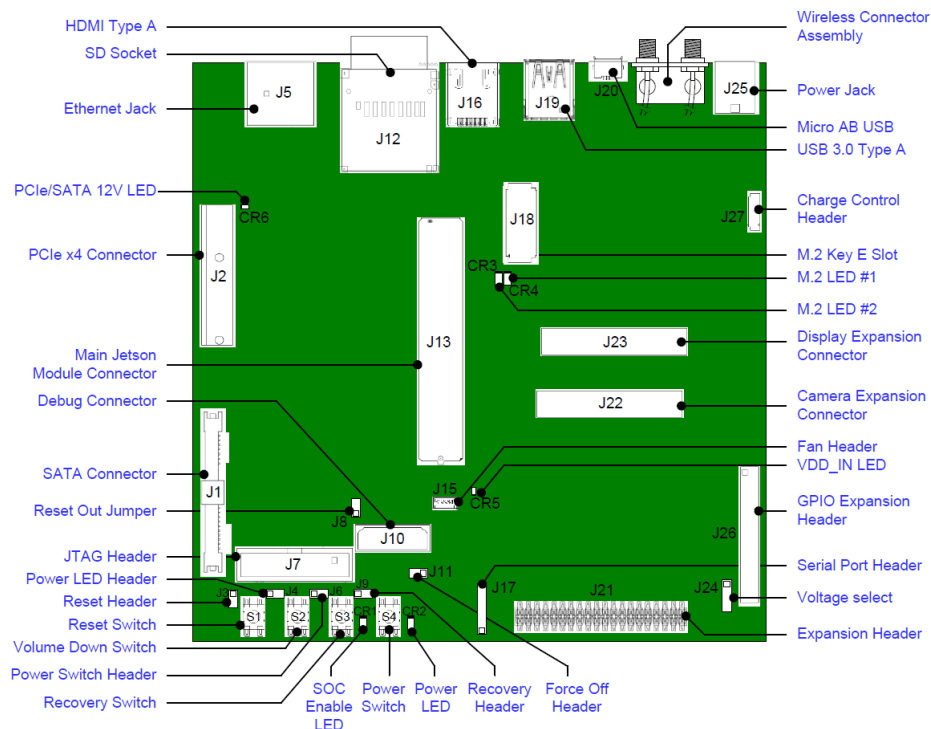


Figura 2: O módulo TX1.



Jetson TX1 J21 Header					
Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
	3.3 VDC Power	1	2	5.0 VDC Power	
	SDA1 General I2C #0 Data 3.3.V, I2C Bus 0	3	4	5.0 VDC Power	
	SCL1 General I2C #0 Data 3.3.V, I2C Bus 0	5	6	GND	
gpio216	GPIO_GCLK AUDIO_MCLK Audio Master Clock (1.8/3.3.V)	7	8	TXD0 UART #0 Transmit	
	GND	9	10	RXD0 UART #0 Receive	

Figura 4: Pinagem parcial do header J21 do TX1.

<https://www.jetsonhacks.com/nvidia-jetson-tx1-j21-header-pinout/>

2.2. DLP4500

O projetor DLP4500 conta com uma tecnologia de sistemas microeletromecânicos, que modulam a luz usando microespelhos digitais. Ele conta com um processador DLPC350, um canhão e algumas entradas e saídas. Sendo a entrada de vídeo DSI e via de comunicação I2C as principais utilizadas pelo Firefly.



Figura 5: O projetor DLP4500

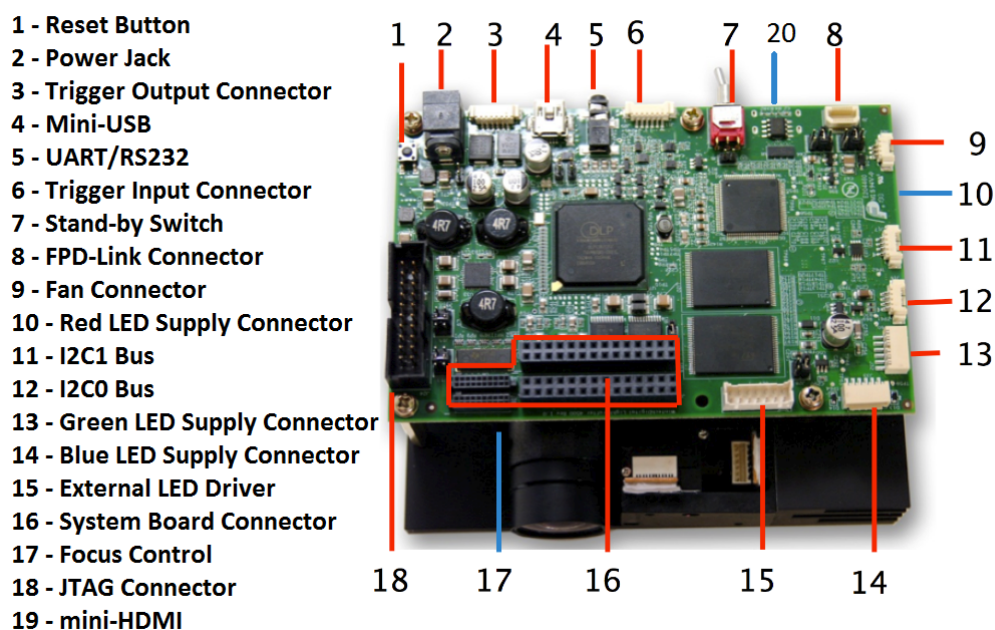


Figura 6: Localização das saídas e entradas no DLP4500.

O projetor precisa de uma fonte 12 V para operar corretamente, conectada no *power jack*, na figura 5 é o conector 2. Existem 2 *bus* I2C, os números 11 e 12; utilizaremos o *bus* 1 (número 11). A conexão é feita por um conector 1.25 mm de quatro terminais, a pinagem é mostrada na tabela 1.

Descrição	Pino	Tensão
I2C SCL	1	3.3 V
I2C SDA	2	3.3 V
VCC de 3.3 V	3	3.3 V
Ground	4	0

Tabela 1: Pinagem dos conectores I2C do DLP4500.

2.3. Bridge

A placa Bridge conta com a conversão DSI-LVDS e com a comunicação I2C entre a TX1 e o DLP4500. A figura 7 mostra a placa Bridge, com a entrada DSI, saídas LVDS A e B, e os pinos de comunicação I2C de 1,8 V e de 3,3 V.

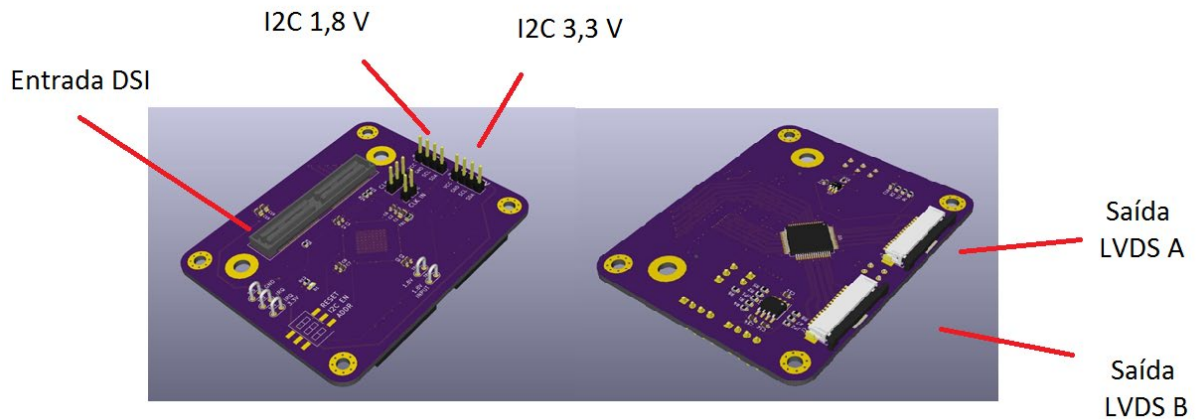


Figura 7: A placa Bridge, vista superior (esquerda) e inferior (direita).

A placa Bridge conta com algumas características para seu funcionamento:

- **Seleção de clock:** O *jumper* de seleção J1 escolhe o tipo de sincronização externa que o chip SN65DSI85 utilizará, podendo ser um *clock* de referência que entra no circuito pelos pinos do J2 (clk_in) ou sem sincronização externa. A figura Figura 11 mostra a ligação para o caso de “clock interno” na chave de seleção CLK_SEL (J1). Caso se deseje ligar na outra posição, o sinal de clock externo deve entrar pelo header J2, conforme figura.

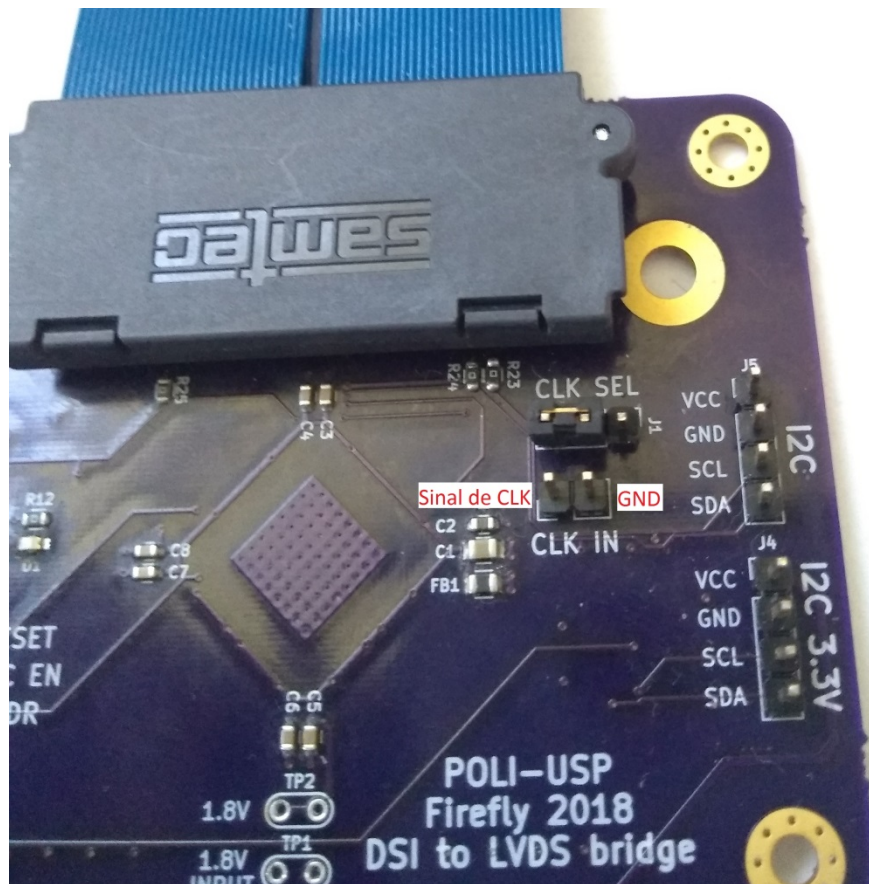


Figura 1 - Detalhe da Bridge: Jumpers do Clock

- **Chave tripla:** A chave de seleção J3 apresenta três *switches*: RESET, I2C EN e ADDR, cada uma com sua função. Vale ressaltar que a chave está ligada ao GND nas três ligações, portanto, quando ela se encontra na posição fechado, ela está fechando o circuito para o ponto terra, logo, nível baixo nas entradas escolhidas. A figura 12 mostra a chave tripla na posição em que todas os *switches* estão em nível baixo.

- RESET: pino que desliga o chip de conversão SN65DSI85 e “reseta” a placa bridge. Ativo em nível baixo, portanto deve se manter na posição aberta (nível alto) para funcionamento da placa.
- I2C EN: pino que habilita a conversão de tensão feita pelo chip TSX0102, que habilita a utilização da comunicação I2C do master com o chip SN65DSI85. Ativo em nível alto, portanto necessário mantê-la em aberto para habilitar o I2C.
- ADDR: endereço de destino da interface I2C local. Para nossa operação, necessário mantê-lo em nível alto, portanto em aberto.

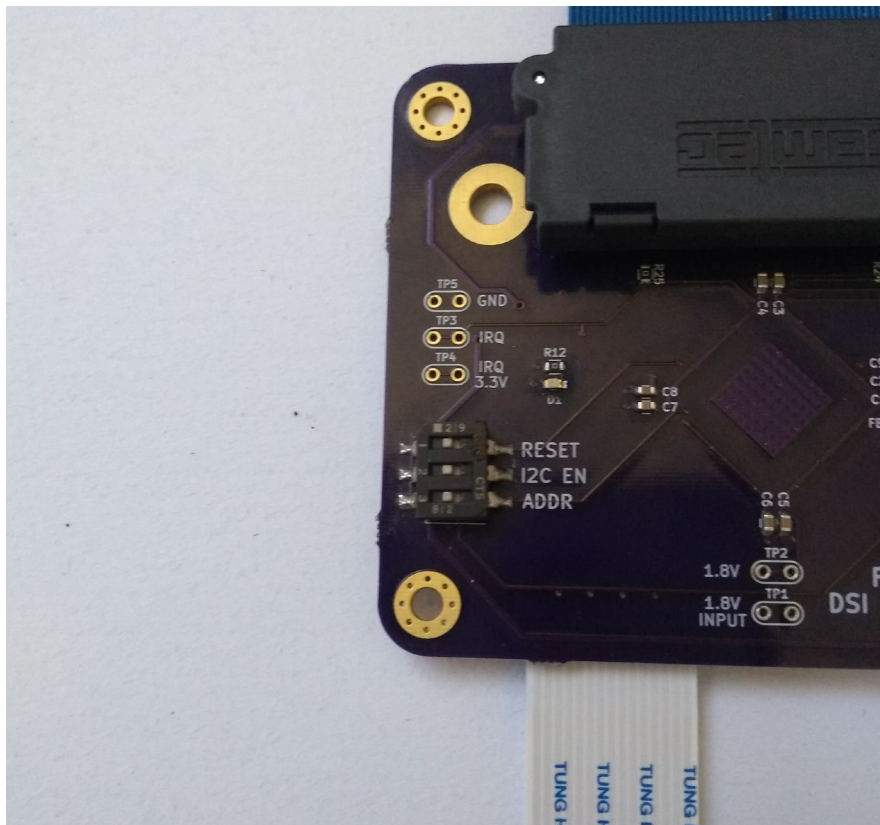


Figura 2 - Detalhe Bridge: Chave tripla

• **Pontos de teste:** Existem alguns pontos de teste no circuito, eles são nomeados TP (*test point*) e servem para auxiliar a comparação dos sinais esperados e reais com uso de osciloscópios ou multímetros, por exemplo. São eles:

- 1V8_IN (TP1): pino de teste que mostra a tensão VCC do header J5 (usualmente, entrada de 1,8 V).
- 1V8 (TP2): pino de teste que mostra a tensão distribuída para os chips TSX0102 (para a entrada de 1,8 V) e SN65DSI85.
- IRQ (TP3): pino de teste que mostra o sinal (de 0 V a 1,8 V) de interrupção que o chip SN65DSI85 envia quando há alguma função programada (via I2C ou de fábrica).
- IRQ_3V3 (TP4): pino de teste que mostra o sinal de interrupção que o chip SN65DSI85 envia, após uma amplificação (de 0 V a 3,3 V).
- GND (TP5): pino de terra, para referência.

3. SOFTWARE

O algoritmo de comunicação foi desenvolvido no ambiente baseado em Linux do sistema operacional da TX1. Sendo necessário um compilador de C++ instalado. Os arquivos estão disponíveis para estudo num diretório do GitHub, o [link](#) se encontra no capítulo de “Arquivos e Links”.

A GPU utilizada foi a NVIDIA Jetson TX1, porém outras podem ser utilizadas, desde que haja comunicação I2C e saída de vídeo DSI, guardadas as devidas alterações. Caso seja a primeira vez que está usando a placa, instale os pacotes necessários para a comunicação I2C. Utilize os seguintes comandos:

```
sudo apt-get update
sudo apt-get install libi2c-dev i2c-tools
```

É necessário habilitar o módulo i2c sempre que a TX1 seja ligada. Para tal, basta fazer:

```
sudo su
modprobe i2c-dev
chmod a+rw /dev/i2c-*
exit
```

Com isso, o módulo está habilitado e pronto para uso. Próximo passo é baixar a pasta com códigos fonte, e extraí-la no diretório escolhido. Abra um terminal e vá até a pasta “Example”. Compile o programa exemplo e rode o programa principal:

```
make
./main
```

Um menu será aberto, com opções de funções para o projetor, todas feitas via I2C. Para acionar uma ação, basta digitar o número dela e pressionar a tecla *enter*. As ações são as seguintes:

Opção	Ação	Descrição
1	Teste Padrão	Mostra um dos testes, a escolha do usuário, de cor e enquadramento do projetor.
2	Teste rotatório	Mostra todos os onze testes de cor e enquadramento, cada um por 1 s, quantas vezes o usuário escolher.
3	Entrada HDMI	Escolhe a entrada HDMI como entrada de vídeo a ser utilizada.
4	Entrada FPD-Link	Escolhe a entrada FPD-Link como entrada de vídeo a ser utilizada.

5	Entrada Flash	Escolhe a imagem salva na memória flash como entrada de vídeo a ser utilizada.
6	Projeção 3D (60 Hz)	Mostra uma demonstração de uma projeção 3D com frequência de 60 Hz.
7	Projeção 3D (120 Hz)	Mostra uma demonstração de uma projeção 3D com frequência de 120 Hz.
10	Mostrar erros	Mostra/esconde as mensagens de erro da comunicação I2C.
0	Sair	Sai do programa, retornando ao terminal.

Tabela 2: Descrição do menu principal do programa que controla o projetor.

4. ARQUIVOS E LINKS

Alguns links para ajudar a compreender cada parte do módulo.

Kit de desenvolvimento da Jetson TX1:

https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/390/JetsonTX1_5F00_TX2_5F00_Developer_5F00_Kit_5F00_Carrier_5F00_Board_5F00_Specification1.pdf

Pinagem completa do *header* J21 do TX1:

<https://www.jetsonhacks.com/nvidia-jetson-tx1-j21-header-pinout/>

Módulo de avaliação DLP4500:

<http://www.ti.com/lit/ug/dlpu011f/dlpu011f.pdf>

Guia de programação do DLP350

<http://www.ti.com/lit/ug/dlpu010g/dlpu010g.pdf>

GitHub com os arquivos:

https://github.com/DiogodosSantos/ResidenciaSW2018_ProjetorInteligente/