LABORATÓRIO TPSE I

Lab 06: Protocolo de Comunicação I2C

Prof. Francisco Helder

6 de novembro de 2023

1 Barramento de Comunicação I2C

O protocolo I2C (Inter-Integrated-Circuit, pronunciado “I squared C” e frequentemente escrito I2C) é para comunicação síncrona entre um mestre e dispositivos escravos usando dois pinos: um pino de dados (SDA) e um de clock (SCL). Frequentemente, o microprocessador é o dispositivo mestre que controla a comunicação com um ou mais dispositivos escravos no barramento. Na BeagleBone, o hardware suporta três barramentos I2C, que são numerados de 0 a 2, e que estão especificados na Tabela 1.

Tabela 1: Configurações para uso dos barramentos I2C disponível na BeagleBone.

HW Bus	**Linux Device**	**Uso padrão**	**Status**
I2C0	/dev/i2c-0	HDMI (Interno da BeagleBone, não tem nos expansores P8 e P9)	Habilitado
I2C1	/dev/i2c-1	SDA pino: P9_18 SCL pino: P9_17	Desabilitado
I2C2	/dev/i2c-2	SDA pino: P9_20 SCL pino: P9_19	Habilitado

Cada chip conectado a um barramento I2C tem um endereço exclusivo que é definido fisicamente no chip. (Às vezes, o projetista de hardware pode selecionar um dos poucos endereços possíveis para um chip.) No caso mais simples, quando o mestre deseja iniciar uma leitura ou gravação em um dispositivo, ele se comunica pelo barramento I2C apropriado e indica o endereço do dispositivo com o qual deseja interagir. Cada dispositivo expõe um conjunto de registradores em um pequeno espaço de endereço. Cada registro tem um propósito especial. Por exemplo, o registro no endereço 0x14 no dispositivo extensor I2C GPIO armazena os 8 bits inferiores que ele irá enviar em seus pinos GPIO. Observe que há três coisas que devem ser especificadas ao interagir com um dispositivo:

1. Em qual barramento um dispositivo está.
2. Qual endereço I2C esse dispositivo possui.
3. De qual endereço de registro ler/gravar (do data-sheet).

2 I2C via Linha de Comando do Linux

Os três barramentos I2C na Beaglebone Black estão endereçados nos seguintes registradores de memória:

- i2c-0: 0x44E0_B000
- i2c-1: 0x4802_A000
- i2c-2: 0x4819_C000

O barramento i2c-0 não é acessível pelos expansores, já os barramento i2c-1 e i2c-2 podem ser utilizados pelos expansores e podem ser utilizados para operações de E/S digital.

2.1 Habilitando o Barramento

Todos os barramentos I2C são controlados através do kernel Linux. Primeiro devemos dizer ao Linux que o barramento I2C de hardware será usado, se ainda não estiver habilitado, seguindo os seguintes passos:

1. Certifique-se se seu sistema tem a ferramenta **i2c-tools**, caso não tenha você deve adicionar esse pacote na sua ferramenta de build e gerar nova imagem do sistema;
2. Determine em qual barramento I2C seu dispositivo está;
3. Verifique o esquema de hardware (ou tabela 1) para determinar qual dispositivo você está acessando. Anote o dispositivo Linux e o endereço;
4. Se você estiver conectando um novo dispositivo I2C, os pinos do expansor P9 da BeagleBone permitem fácil acesso a dois barramentos I2C: I2C1 e I2C2. (I2C0 é interno ao BeagleBone.)
   - O barramento I2C1 tem SDA em P9_18 e SCL em P9_17.
   - O barramento I2C2 tem SDA em P9_20 e SCL em P9_19.
5. Display which I2C buses Linux currently has enabled:

```
$ i2cdetect -l
i2c-1        i2c        OMAP I2C adapter        I2C adapter
i2c-2        i2c        OMAP I2C adapter        I2C adapter
i2c-0        i2c        OMAP I2C adapter        I2C adapter
```

6. Se o seu dispositivo estiver no barramento de hardware I2C1, talvez você precise primeiro habilitar o suporte do Linux para o barramento (/dev/i2c-1). Cheque se os pinos estão configurado para i2c:

```
$ config-pin -q P9_19
$ config-pin -q P9_20
```

   Se imprimir no modo “i2c”, então está configurado. Caso contrário, quando você não tiver capes carregadas, o cape universal pode estar ativa, o que torna os dois pinos para I2C-1 disponíveis como GPIO. Você deve alterar a configuração dos pinos de GPIO para I2C:

```
$ config-pin P9_19 i2c
$ config-pin P9_20 i2c
```

7. Exiba os dispositivos I2C no barramento I2C escolhido:

```
$ i2cdetect -y -r 2
```

   - onde **2** refere-se ao dispositivo Linux **/dev/i2c-1**
   - Saída será:

```
$ i2cdetect -y -r 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- 3c -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

"–" significa que não tem dispositivo.
"XX" significa que o dispositivo foi detectado no endereço XX (hex)
"UU" em /dev/i2c-1 (HW I2C1) significa em uso por um driver (HDMI ou outro driver de dispositivo do kernel).

8. Você deve definir a configuração de pinos toda vez que o BeagleBone for reinicializado.

   - Se você estiver usando qualquer um dos dispositivos I2C em I2C-1, provavelmente precisará que seu programa tente configurar os pinos P9_19 e P9_20 para operação I2C usando os comandos config-pin acima.

9. Soluções de problema:

   - se você executar o comando:

     ```
     $ i2cdetect -y -r 2
     ```

     e leva muito tempo (segundos por endereço), e não encontra nada, então você provavelmente precisará alterar seus pinos para usar I2C

     – Mude as configurações dos pinos pat I2C:

       ```
       $ config-pin P9_19 i2c
       $ config-pin P9_20 i2c
       ```

     – Você pode vê atual configuração:

       ```
       $ config-pin -q P9_19
       $ config-pin -q P9_20
       ```

     – Pode vê se o pino está disponível:

       ```
       $ config-pin -l P9_19
       $ config-pin -l P9_20
       ```

2.2 Comunicando com Dispositivo via I2C

Exiba a memória interna de um dispositivo I2C:

```
$ i2cdump -y 2 0x3c
No size specified (using byte-data access)
     0 1 2 3 4 5 6 7 8 9 a b c d    0123456789abc
00: ff ff 00 00 00 00 00 00 00 00    ..............
10: 00 00 00 00 00 00 00 00 00 00    ..............
...
```

Isso mostra a memória interna do dispositivo no endereço 0x3c (expansor) em /dev/i2c-2 (HW I2C2). A saída pode ser diferente para você.

- Consulte a folha de dados do seu dispositivo I2C para identificar o que significa cada endereço de registro.
- Você também pode ler um único byte de memória, se desejar:

```
$ i2cget -y 2 0x3c 0x00
```

- Arqumentos:
  - **-y**: desativa o prompt de confirmação;
  - **2**: barramento I2C /dev/i2c-2;
  - **0x3c**: Endereço do dispositivo no barramento;
  - **0x00**: Endereço do registrador para leitura.

Escreva no dispositivo I2C usando o comando **i2cset**:

```
$ i2cset -y 2 0x3c 0x00 0x00
$ i2cset -y 2 0x3c 0x01 0x00
```

- Estes comandos controlam o dispositivo com endereço 0x3c em /dev/i2c-2: no registrador 0x00 escreve 0x00, e no registrador 0x01 escreve 0x00.
- Lembrando que você deve garantir que o pino no expansor dever estar configurado para i2c.

Memoria interna do dispositivo:

```
$ i2cdump -y 2 0x3c
No size specified (using byte-data access)
     0 1 2 3 4 5 6 7 8 9 a b c d        0123456789abc
00: 00 00 00 00 00 00 00 00 00 00       ..............
10: 00 00 00 00 00 00 00 00 00 00       ..............
...
```

A saída pode ser diferente; espere ver os dois primeiros valores (em 0x00 e 0x01) ambos definidos como 0 agora.

3 Atividades Práticas

pratica 1

Escolha um dispositivo qualquer que se comunique via I2C e realize os testes acima.

pratica 2

Agora crie uma aplicação (na sua linguagem favorita) que se comunique com esse dispositivo, inicialize esse dispositivo e coloque em funcionamento.