

LABORATÓRIO AOC II



Lab 01: Primeiro Programa Embarcado GPIO

Prof. Thiago Werley

7 de novembro de 2024

1 Hello, World!

- Parece que todo livro de programação já escrito começa com o mesmo exemplo, um programa que imprime “**Hello, World!**” na tela do usuário.
- A mensagem “**Hello, World!**” serve como uma referência útil para usuários de linguagens de programação e plataformas de computador.
- A maioria dos sistemas embarcados não possui um monitor ou dispositivo de saída análogo.
- Seria muito melhor começar com um programa embarcado pequeno, de fácil implementação e altamente portátil, no qual houvesse pouco espaço para erros de programação.
- O Programador embarcado deve sempre começar cada novo projeto com a suposição de que nada funciona, que tudo em que podem confiar é a sintaxe básica de sua linguagem de programação.

1.1 Programa piscar LED

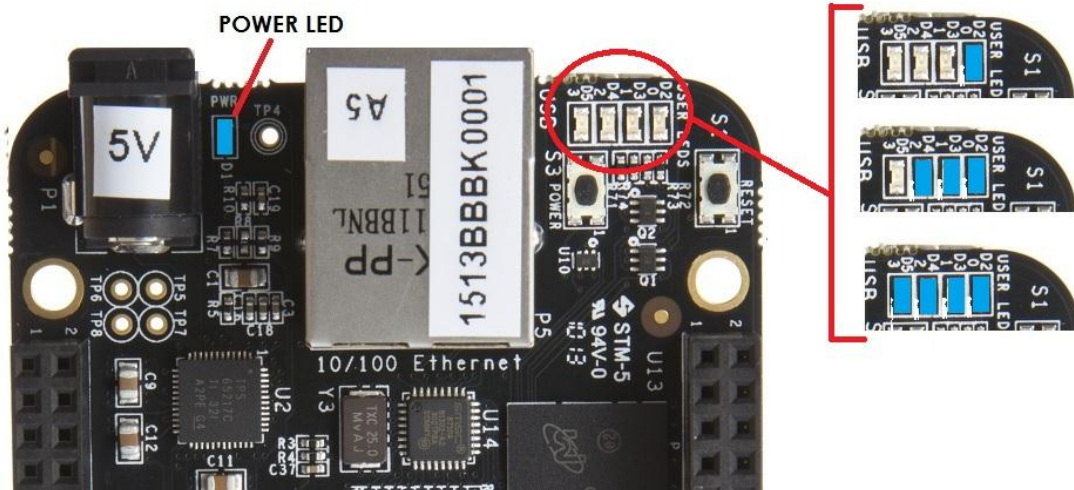
- Quase todos os sistemas embarcados que encontramos em nossas respectivas carreiras têm pelo menos um B que poderia ser controlado por software.
- Um substituto popular para “**Hello, World!**” programa é aquele que pisca um LED em uma taxa de X Hz (um ciclo liga-desliga completo por segundo).
- O código necessário para ligar e desligar um LED é limitado a algumas linhas de código, portanto há muito pouco espaço para a ocorrência de erros de programação.
- Nosso primeiro passo é aprender como controlar o LED verde que queremos alternar.

1.2 Programa piscar LED

- O **LED azul** está identificado como “**LED2**” no módulo adicional.
- O **Manual de Referência do Sistema BeagleBone Black** e o **Manual de Referência Técnica dos Processadores AM335x Sitara™** descrevem como os LEDs do módulo adicional são conectados ao processador.
- Os esquemas também podem ser usados para rastrear a conexão do LED até o processador, que normalmente é o método que você precisa usar quando tiver seu próprio hardware.

1.3 Programa piscar LED

GPIO1_21	V15
GPIO1_22	U15
GPIO1_23	T15
GPIO1_24	V16



- A seção de E/S de uso geral do Manual de referência técnica dos processadores AM335x Sitara™ mostra que o sinal OUT é controlado pelo **pino 23** do GPIO do processador.
- Portanto, precisaremos ser capazes de definir o **pino 23** do GPIO alternadamente como alto e baixo para que nosso programa de pisca-pisca funcione corretamente.
- A superestrutura do programa LED Piscando é mostrada a seguir. Esta parte do programa é independente de hardware.

2 Atividades em Sala

```
#include "hw_types.h"
#include "soc_AM335x.h"

/*****
**      INTERNAL MACRO DEFINITIONS
*****/
#define TIME      1000000
#define TOGGLE    (0x01u)

#define CM_PER_GPIO1 0xAC
#define CM_PER_GPIO1_CLKCTRL_MODULEMODE_ENABLE (0x2u)
#define CM_PER_GPIO1_CLKCTRL_OPTFCLKEN_GPIO_1_GDBCLK (0x00040000u)

#define CM_conf_gpmmc_ben1 0x0878
```

```
#define GPIO_OE                                0x134
#define GPIO_CLEARDATAOUT                      0x190
#define GPIO_SETDATAOUT                       0x194

int _main(void) {
    // ini flag for blink LED
    unsigned char flagBlink=0;                //init flag
    unsigned int val_temp;
    volatile unsigned int ra;

    // Configure the green LED control pin
    /*****
    * configure clock GPIO in clock module
    *****/
    HWREG(SOC_CM_PER_REGS+CM_PER_GPIO1) |=
        CM_PER_GPIO1_CLKCTRL_OPTFCLKEN_GPIO_1_GDBCLK |
        CM_PER_GPIO1_CLKCTRL_MODULEMODE_ENABLE;

    /*****
    * configure mux pin in control module
    *****/
    HWREG(SOC_CONTROL_REGS+CM_conf_gpmc_a5) |= 7;

    /*****
    * set pin direction
    *****/
    val_temp = HWREG(SOC_GPIO_1_REGS+GPIO_OE);
    val_temp &= ~(1<<21);
    HWREG(SOC_GPIO_1_REGS+GPIO_OE) = val_temp;

    while (1){
        // Change the state of the green LED.
        flagBlink ^= TOGGLE;

        if(flagBlink){
            HWREG(SOC_GPIO_1_REGS+GPIO_SETDATAOUT) =
                1<<21;
        }else{
            HWREG(SOC_GPIO_1_REGS+GPIO_CLEARDATAOUT) =
                1<<21;
        }
        // delay
        for(ra = 0; ra < TIME; ra ++);
    }
    return(0);
} /* ----- end of function main ----- */
```

3 Atividades Práticas

pratica 1:

Crie um sistema no qual pisca mais de um LED.

pratica 2:

Crie um sistema no qual a sequência de pisca dos quatro LEDs internos mude de 0 até 15.