

Sistemas Embarcados I - Laboratório 9
Guilherme Goes Zanetti 2019107824 / Luiza Batista Laquini 2019107786
02/03/2022

Objetivos

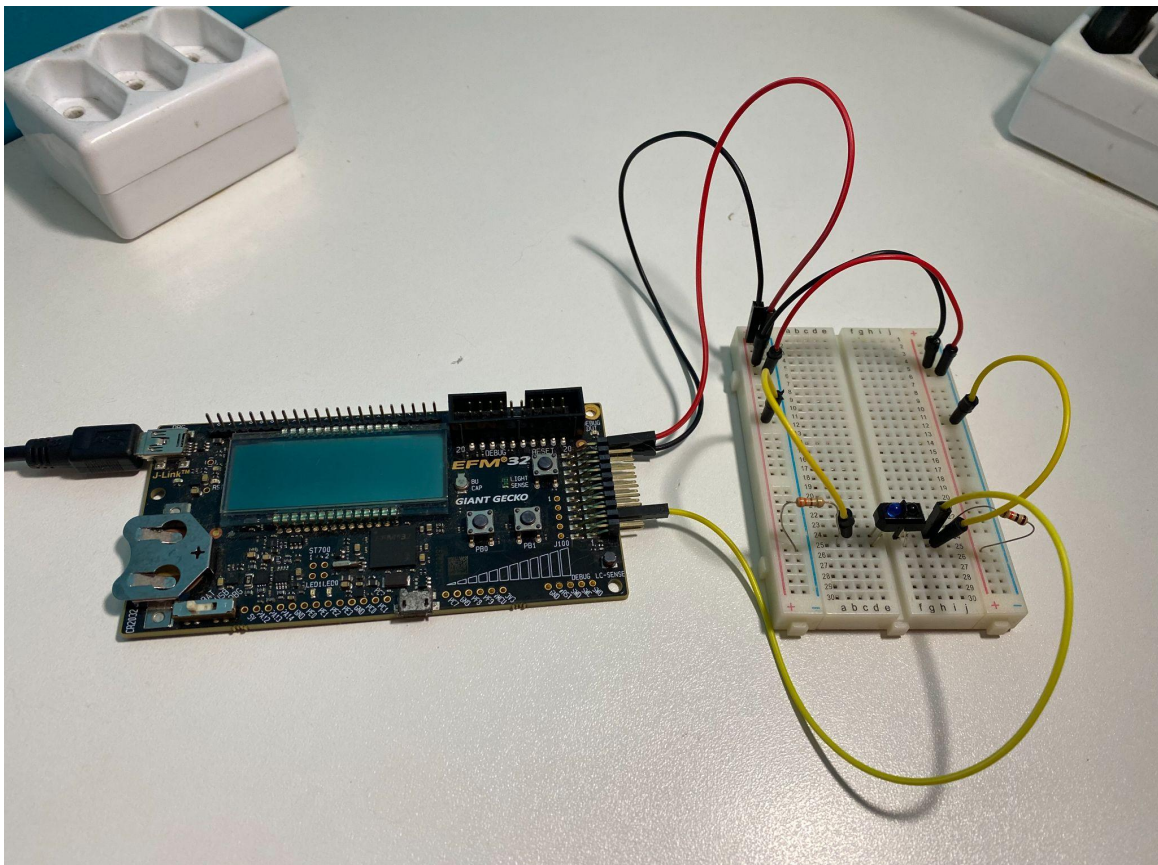
Utilizando a EFM32 Giant Gecko disponibilizada pelo professor:

- Utilizar as funcionalidades de comunicação serial da placa de desenvolvimento;
- Reforçar conceitos anteriores como utilização de PWM e leitura ADC.

Resultados

Primeiramente, para realizar o experimento, inicializamos as portas que utilizamos, o ADC para a leitura do sensor, o UART e o PWM para regular a intensidade do LED0. Tudo isso no início da main.

O circuito físico foi reutilizado do último laboratório, tendo um sensor óptico reflexivo TCRT 5000, como mostrado na foto a seguir:



Em seguida foi utilizada a biblioteca de UART para enviar e receber caracteres de um monitor serial do Arduino IDE, opção mais simples para nosso caso. Assim, conseguimos receber os caracteres sequencialmente para montar uma instrução e executá-la de acordo.

As instruções que podem ser recebidas pelo UART foram as seguintes:

- **LED1_ON e LED1_OFF** - Ligam ou desligam o LED 1;
- **LED0_0, LED0_25, LED0_50, LED0_75 e LED0_100** - Definem o valor do PWM para o LED0, sendo que 50 significa 50% do total e assim por diante;

A segunda parte do laboratório foi a exibição do estado dos dois botões inseridos na placa e da leitura ADC do sensor de reflexão. Para isso, exibimos a cada certa quantidade de loops - para não imprimir informação demais no monitor serial - o estado dos dois botões e a tensão sendo lida do sensor. Ou seja, exibimos se os botões estão pressionados ou não e também o valor de tensão do sensor convertido em milivolts.

```
106 void readPrintButtons(){
107     int buttons = GPIO_ReadPins(GPIOB);
108
109     UART_SendString("\n\n/////////\n");
110
111     if((buttons & BIT(9)) > 0){
112         UART_SendString("Button PB0 is NOT pressed\n");
113     } else {
114         UART_SendString("Button PB0 is pressed\n");
115     }
116
117     if((buttons & BIT(10)) > 0){
118         UART_SendString("Button PB1 is NOT pressed\n");
119     } else {
120         UART_SendString("Button PB1 is pressed\n");
121     }
122 }
123
124 void printAnalogRead(){
125     char buffer[100];
126     int value = ADC_Read(ADC_CH0);
127
128     value = 1000 * 3.3 * value / 4095.0; //Convert from 2^12 to volts
129
130     UART_SendString("Sensor read is: ");
131     UART_SendString(itoa(value, buffer, 10));
132     UART_SendString(" mV\n");
133 }
```

Segue vídeo apresentando os resultados alcançados: [LAB9_Video](#)

Conclusão

Conseguimos, por meio desse experimento, compreender melhor o funcionamento da biblioteca de UART fornecida. Essa layer de abstração facilita em

muito a utilização de comunicação serial, que se torna quase tão simples de utilizar quanto um `printf()` ou `scanf()` no C convencional.