

# LPC1769EXPRESSO – Comunicação UART com Sensores de Temperatura e Luminosidade

Gustavo Henrique Santana de Souza  
Ney Correa da Silva Neto

Inatel - Instituto Nacional de Telecomunicações  
2017

## Sensores

---

### Sensor de Luminosidade ISL29003

O ISL29003 é um sensor de luz integrado com um sensor de 16 bits. Em funcionamento normal, o consumo de energia é inferior a 300µA. Projetado para operar com tensões de 2.5V a 3.3V, especificado para operação de -40 ° C a + 85 ° C.

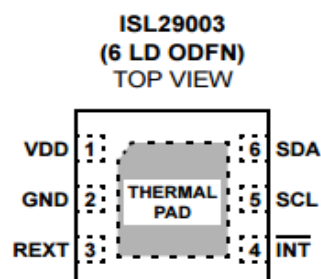
#### Aplicações:

- Detecção de Luz Ambiente
- Controle de luz de Fundo
- Medidores de luz de câmeras
- Controles de Iluminação
- Sistemas de controle de temperatura

#### Pinos Utilizados:

- PIN 6: SDA I2C serial data

#### Configuração dos Pinos:



## Sensor de Temperatura **MAX6576/MAX6577**

O MAX6576 / MAX6577 Possui saída de um unico fio. Converte a temperatura ambiente em uma onda quadrada com uma frequência proporcional à temperatura. Oferece uma precisão de  $\pm 3^{\circ}\text{C}$  em  $+ 25^{\circ}\text{C}$ ,  $\pm 4,5^{\circ}\text{C}$  a  $+ 85^{\circ}\text{C}$  e  $\pm 5^{\circ}\text{C}$  a  $+ 125^{\circ}\text{C}$ . O intervalo período / frequência da saída Onda quadrada pode ser selecionada por dois pinos (TS0, TS1) para VDD ou GND. Disponíveis em 6 pinos SOT23.

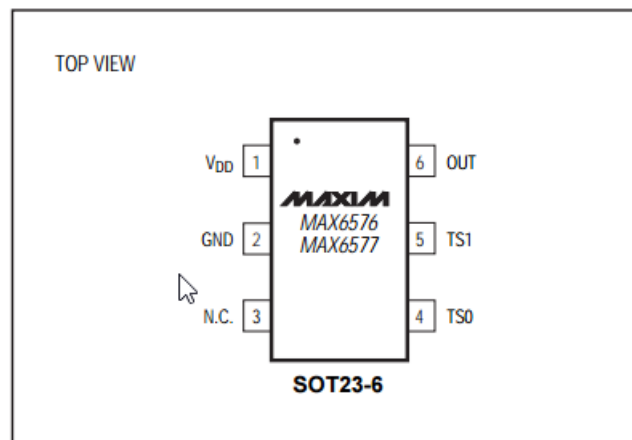
### *Aplicações:*

- Monitoramento crítico de temperatura  $\mu\text{P}$  e  $\mu\text{C}$
- Equipamentos portáteis alimentados por baterias
- Celulares
- Baterias
- Discos rígidos / unidades de fita
- Equipamentos de rede e telecomunicações
- Equipamento médico
- Automotivo

### *Pinos Utilizados:*

- TS1, TS0

### *Configuração dos Pinos:*

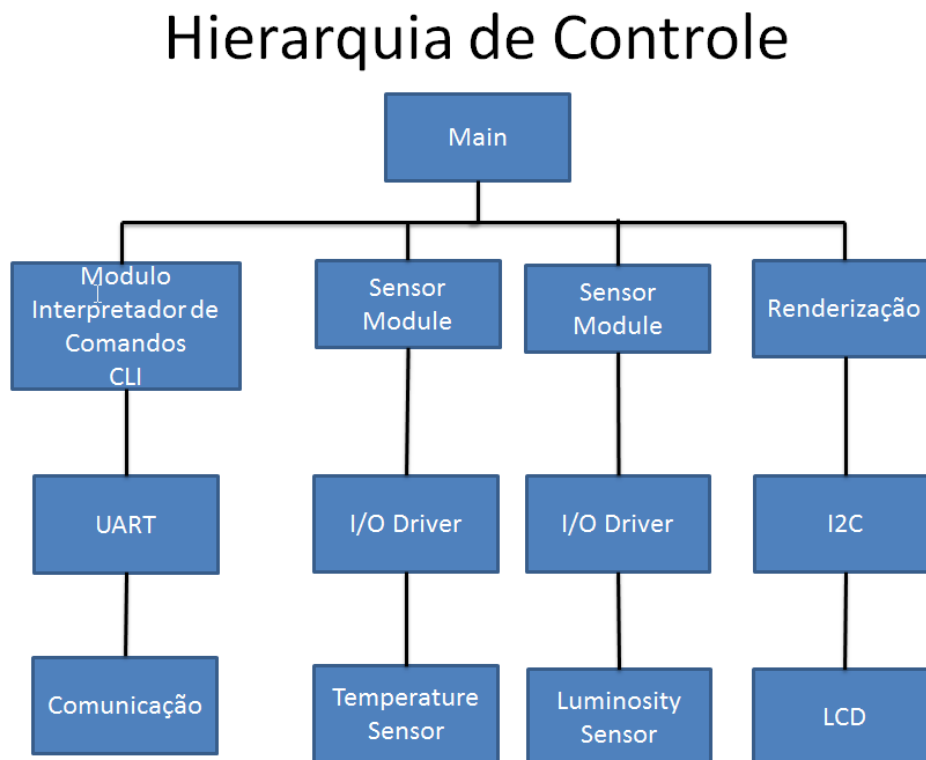


# Diagramas

---

## Hierarquia de Controle

---



## Diagrama em Camadas

---

# Camadas

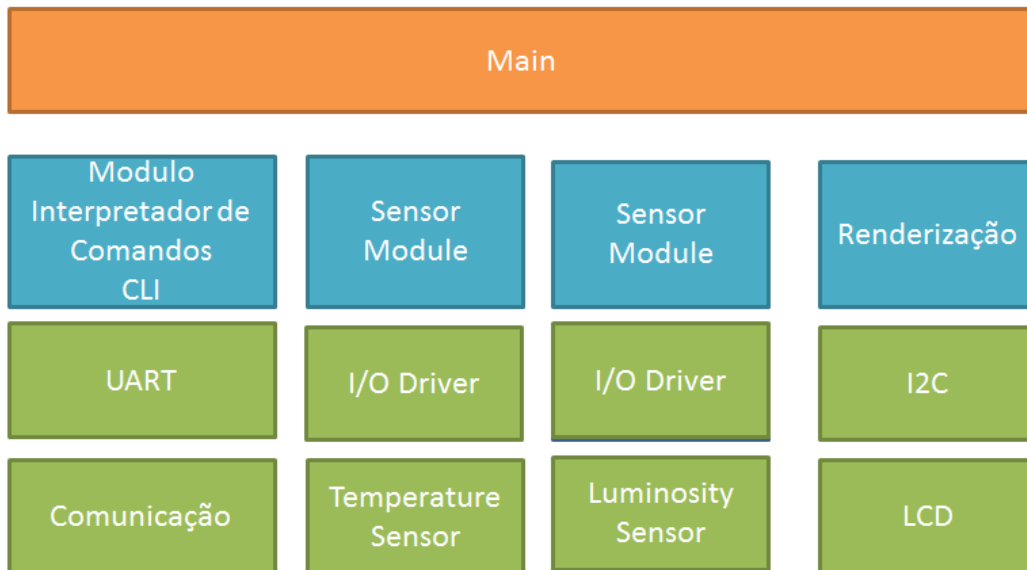
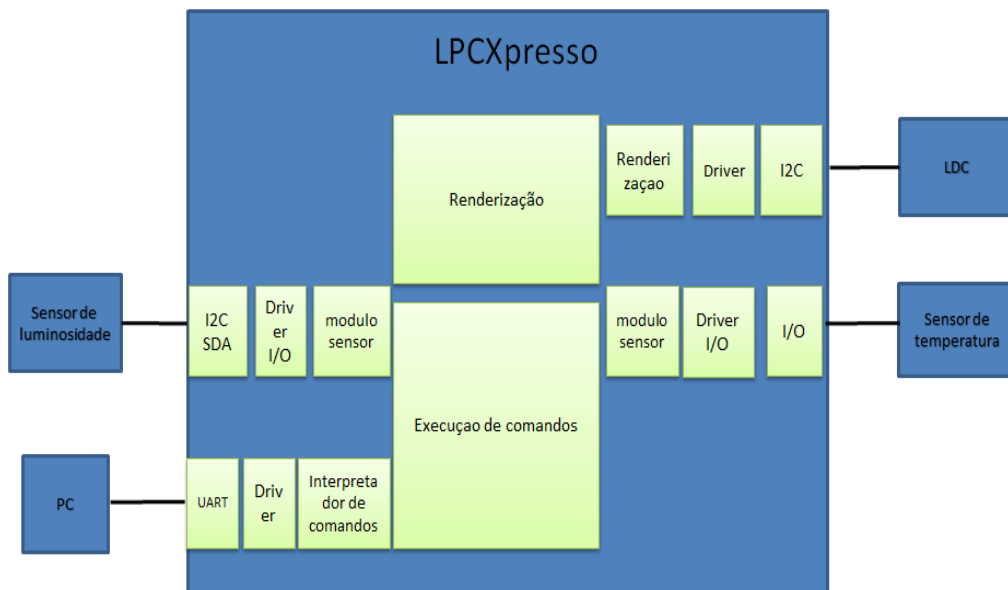


Diagrama em Blocos

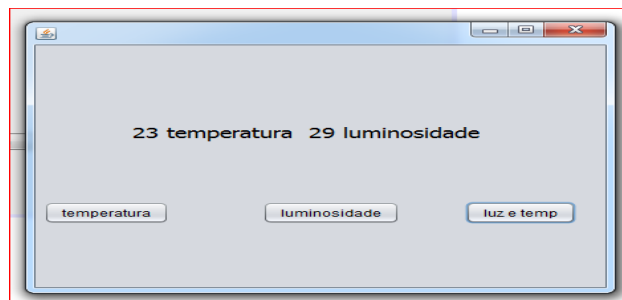
## Diagrama em Blocos

Projeto EC020

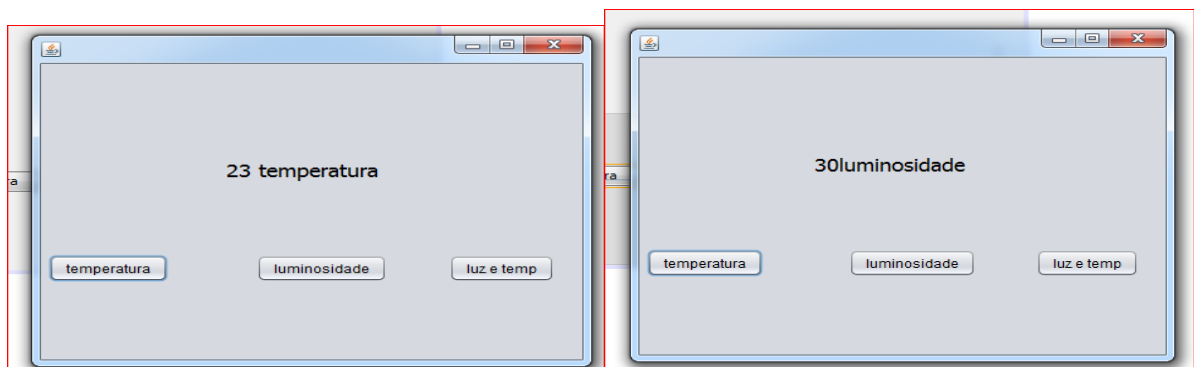


# Funcionamento

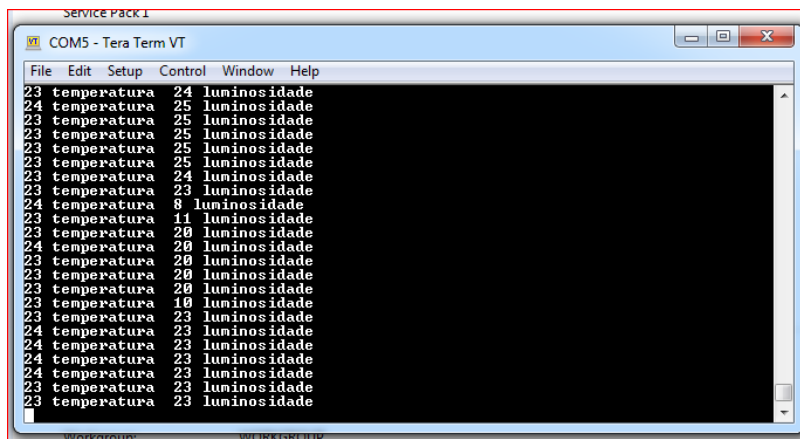
O projeto consiste informar ao usuário a informação de Luminosidade e Temperatura Ambiente através de um Software Java e Display. Para obter tal informação basta clicar na opção desejada através da tela do Software JAVA:



O valor desejado será mostrado de acordo com o botão clicado. Este comando é enviado via UART para a placa LPC1769EXPRESSO e recebida informação no Software JAVA.



Também poderá ser Enviada/Lida através do software TERATERM:



# Software

---

## Descrição Técnica

Consiste no uso de UART para envio e recepção de variáveis de decisão para os sensores de temperatura e luminosidade. A leitura será feita através de 2(dois) sensores (temperatura e luminosidade) e retornados para o Display seus respectivos valores de leitura. A Uart servirá para efetuar o comando de entrada no menu SWITCH CASE elaborado no código.

## Funções

### **UART\_SENSTRING()**

Envia o valor das variáveis correspondente a temperatura e luminosidade para o PC através da comunicação serial

### **UART\_RECEIVE()**

Lê a variável de decisão que é recebida através da comunicação serial

### **SPRINTF()**

Transformar variável inteira em char.

### **INIT\_UART3()**

Inicializa a uart.

### **INIT\_I2C()**

Inicializa a entrada i2c.

### **OLED\_INIT()**

Inicializa o display

## Variáveis

### **VARIÁVEL GLOBAL :**

Variável **len** recebe os valores de entrada UART que entram na estrutura de repetição SWITCH CASE

#### VARIÁVEL LOCAL :

- **data** = valor da uart.
- **t** = valor do sensor.
- **t2** = valor do sensor em célsius.
- **lux** = valor do sensor de luminosidade.
- **str** = valor da temperatura em char.
- **str2** = valor da luminosidade em char.

### Estrutura

#### CASE: 0

Envia a temperatura no display e na serial.

#### CASE: 1

Envia a luminosidade no display e na serial.

#### CASE: 2

Envia enviar a temperatura e luminosidade no display e na serial.

#### DEFAULT:

Retorna ao inicio do laço de repetição.

## Utilizando RTOS

---

Para uma segunda versão do projeto foi utilizado o sistema operacional em tempo real, onde o foco foi substituir o modo que as funções principais de leituras de sensores de temperatura e luminosidade fossem executados. Anteriormente todas as funções de leitura e lógica de projeto eram executadas dentro da função principal Main dentro de loops de repetição (while), com as alterações para o sistema operacional em tempo real, foram criadas TASKS para que que fossem responsáveis pela leitura individual de cada sensor e da execução lógica do programa.

### Fila/buffers

```
xQueueHandle xQueue;
static uint8_t buf[10];
static uint8_t buf2[10];
uint32_t len;

int main( void )
{
    /* The queue is created to hold a maximum of 5 long values. */
    xQueue = xQueueCreate( 5, sizeof( uint8_t ) );
```

Para a leitura de variáveis enviadas pela serial UART foi declada uma fila xQueue, e respectivamente para armazenar a leitura dos sensores de temperatura e luminosidade os buffers buf e buf2.

## Criação das Tasks

```
xTaskCreate( vSenderTask, "Sender1", 240, ( void * ) 100, 1, NULL );  
xTaskCreate( vUartTask, "Sender2", 240, ( void * ) 200, 1, NULL );  
xTaskCreate( vSensorTemptask, "Sender2", 240, ( void * ) 200, 1, NULL );  
xTaskCreate( vSensorLumitask, "Sender2", 240, ( void * ) 200, 1, NULL );
```

Através da função xTaskCreate, é criada cada Task, onde é associada a um Nome “Sender1” e “Sender2”, e também é passado como parâmetro seu valor de prioridade “1”.

## Task de Temperatura

```
static void vSensorTemptask( void *pvParameters )  
{  
  
    int32_t t = 0;  
    uint32_t t2 = 0;  
  
    for(;;){  
        vPrintString( "vSensorTemptask.\r\n" );  
        t = temp_read();  
        t2 = (t*26)/257;  
        intToString(t2, buf, 10, 10);  
        vTaskDelay( 100 / portTICK_RATE_MS );  
  
        //taskYIELD();  
    }  
}
```

Através da função xSensorTemptask, é feita a leitura de temperatura do sensor, e enviada para o buffer “buf”.

## Task de Luminosidade

```
static void vSensorLumitask( void *pvParameters )  
{  
    uint32_t lux = 0;  
  
    for(;;){  
        vPrintString( "vSensorLumitask.\r\n" );  
  
        lux = light_read();  
        intToString(lux, buf2, 10, 10);  
        vTaskDelay( 100 / portTICK_RATE_MS );  
  
        //taskYIELD();  
    }  
}
```

Através da função xSensorLumitask, é feita a leitura de luminosidade do sensor, e enviada para o buffer “buf2”.

## vSenserTask

Nesta task foi atribuído os cases citados anteriormente para o funcionamento lógico do programa. Onde é feita a leitura do buffer para mostrar no Display os valores dos sensores, e executar os comandos que foram atribuídos na fila xQueue.



### Task de Leitura UART

```
static void vUartTask( void *pvParameters )
{
    uint8_t data;
    portBASE_TYPE xStatus;

    for(;;){
        vPrintString( "vUartTask.\r\n" );
        if(UART_Receive(UART_DEV, &data, 1, NONE_BLOCKING) >0){
            xStatus = xQueueSendToBack( xQueue, &data, 0 );
            if( xStatus != pdPASS )
            {
                /* We could not write to the queue because it was full □ this must
                be an error as the queue should never contain more than one item! */
                vPrintString( "Could not send to the queue.\r\n" );
            }
        }
        vTaskDelay( 50 / portTICK_RATE_MS );
    }
}
```

A função vUartTask faz a leitura de variáveis do teclado enviadas pela serial UART. Esta variável é armazenada na fila xQueue como data, e posteriormente utilizada para a execução lógica do programa, ou seja, para enviar comandos do computador para a placa LPCX.

## Resultados práticos

---

O software apresentou consistência em relação ao uso da SERIAL utilizando a interface UART para a execução de comandos, e também na leitura dos sensores de temperatura e luminosidade que estão acoplados na placa LPCEXPRESSO1769. O conhecimento sobre o uso de interfaces foi essencial para a execução de leitura dos sensores utilizados no projeto. O uso de muitas bibliotecas para o funcionamento acarretou certo impedimento de produtividade, pois apresenta diversos erros de compilação que conseqüentemente podem acarretar atrasos no projeto. De certa forma, este trabalho explorou bem o uso de ferramentas (datasheet) do fornecedor, e metodologias de padrão de projeto.

### Informação no Display OLED

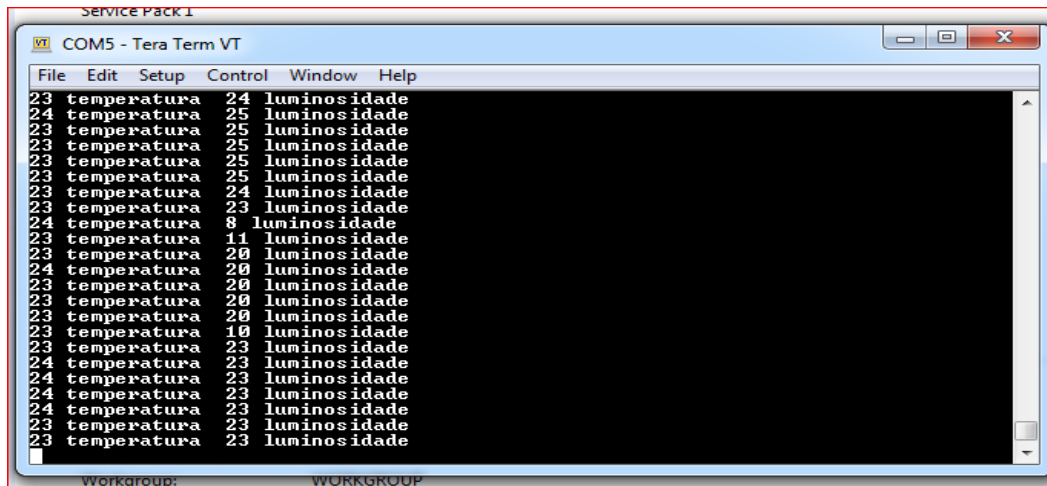
---



Informação de Leitura de Luminosidade e Temperatura mostrada no display OLED

#### *Informação TERATERM*

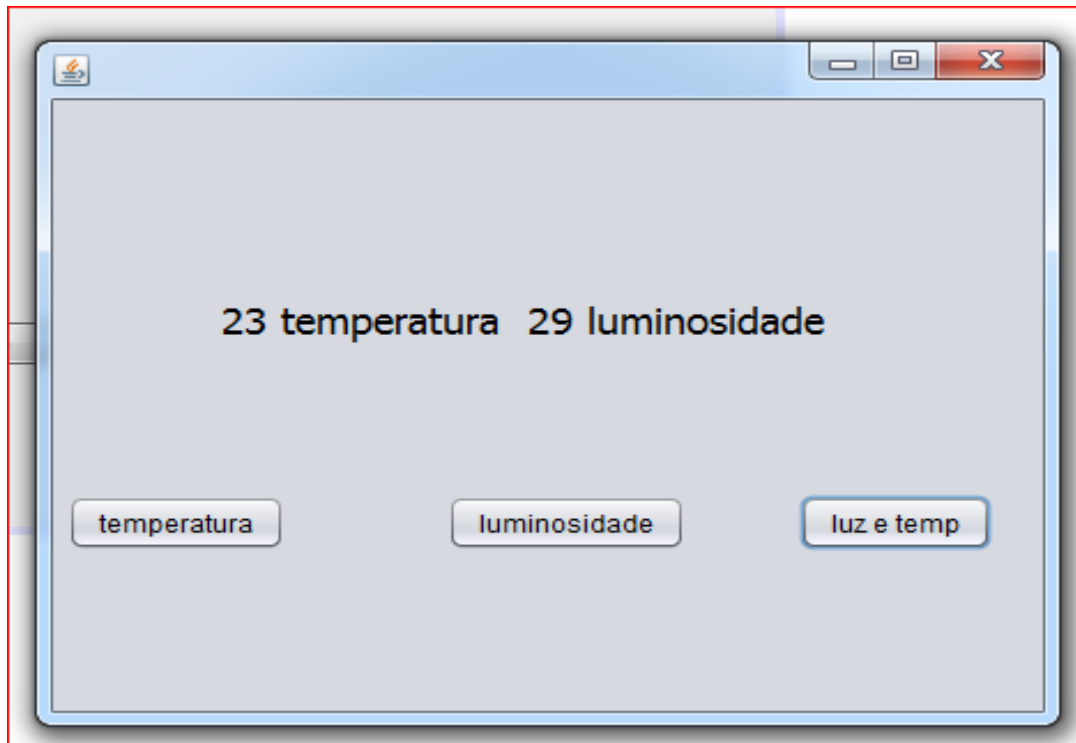
---



Informação mostrada pelo TERATERM via UART

#### *Informação JAVA*

---



A informação pode ser mostrada via Software JAVA, alternando entre luminosidade e temperatura.