

Projeto Final  
Laboratório de sistemas microprocessados



David Herbert de Souza Brito - 200057405  
Gabriel Cruz Vaz Santos - 200049038

## 1. Objetivo do trabalho e proposta de resolução

O Trabalho tem como objetivo a obtenção do valor atual do bitcoin em dólar para exposição em real, com as conversões e atualizações automáticas. Para resolução, o valor do bitcoin é chamado via API para sua exposição em um lcd.

Os componentes necessários para implementação são:

- MSP 430F5529
- Esp 32
- Monitor LCD 1602A

## 2. Implementação e configurações do projeto

Foi utilizado o protocolo de comunicação UART ( Universal Asynchronous Receiver/Transmitter ) para a comunicação entre o MSP430 e o esp32. Para o primeiro, foram utilizados os pinos P3.3 e P3.4 para receptor e transmissor, respectivamente, enquanto que para o segundo foram utilizados os pinos RX2 ( GPIO 16 ) e TX2 ( GPIO 17 ).

Para a conexão entre o MSP e o lcd, foi utilizado o protocolo i2c ( Inter-Integrated Circuit ). Logo, para o controlador da Texas foi-se utilizado os pinos P3.0 e P3.1 para conectar aos pinos SDA ( envio de dados) e SCLK ( controle do clock ), respectivamente. Toda configuração foi feita através do Code Composer Studio com uso das máscaras disponibilizadas pela própria empresa em sua documentação. Para esse caso, não foi necessário o uso de interrupções, apenas trava de execução para aguardar a liberação do buffer e assim permitir a escrita no periférico.

Enquanto isso, o microcontrolador da *espressif* é configurado em dois locais diferentes. Primeiro, é necessário ir ao CCS ( code composer Studio ) e preparar o MSP430 para comunicação UART. Para tal, foi-se utilizado interrupções, tendo em vista que o dispositivo não sabe quando receberá a informação do bitcoin.

A configuração da esp32 propriamente dita pode ser feita tanto usando dos drivers Esp-idf ou a interface do Arduino. Optou-se por utilizar os drivers fornecidos pela *espressif*, pois permitem mais controle do código.

Utilizou-se do módulo Wi-Fi para fazer a chamada API a fim de obter o valor do bitcoin. Assim que esse valor é obtido, é feito sua conversão e o envio via UART para o MSP430, responsável por imprimir o valor no lcd. No entanto, apesar do sucesso ao configurar wifi, devido à questões de segurança ( explicação na seção 3 ), não foi possível atualizar o valor automaticamente.

O MSP430 só recebe um byte por vez, por isso na configuração do ESP32 optou-se por enviar um byte por vez do que será apresentado no lcd, que é guardado em um buffer interno e somente ao final apresentado. O sucesso dessa implementação deu-se ao *envio manual* de um **acknowledge** de volta ao esp após o controlador da Texas entrar em sua trava de execução.

### **3. Análise do resultado**

Durante o desenvolvimento do projeto, ocorreram-se muitos erros devido a instalar as dependências necessárias para o esp32 e também problemas de hardware da própria placa. Apesar dos drivers instalados corretamente e o seu devido reconhecimento no computador, era impossível dar flash no dispositivo, o que impediu a leitura e escrita de dados. Após muitos testes diferentes, foi-se descoberto que a causa estava na própria placa e foi necessário trocá-la.

Após isso, a comunicação UART foi um sucesso, no entanto, não pode-se dizer o mesmo da requisição http, que devido a questões de segurança, não foi possível fazer a chamada. A conexão com wifi foi um sucesso, mas a atualização do esp-idf após o 5.1 fez a chamada http ser muito mais custosa do que deveria.

### **4. Conclusão e projetos futuros**

Para futuro desenvolvimento, o ideal seria conseguir fazer a requisição http do valor atual do bitcoin, além não só adicionar um Buzzer para indicar que foi feita uma atualização no valor lido, mas também Led's de diferentes cores para indicar o tipo de mudança. Por exemplo:

- Led vermelho caso tenha ocorrido uma queda de mais de 10% em relação ao último valor;
- Led amarelo para quedas menores de 10%
- Led verde para aumentos de valor

### **5. Links**

- github: <https://github.com/Leir-Cruz/sismic-bitcoinRating>
- vídeo no Youtube demonstração: <https://www.youtube.com/watch?v=WMyhbpO5L9o>

