

Universidade de Brasília - UnB

Faculdade do Gama - FGA

Eletrônica Embarcada 2/2019

Prof. Drº Gilmar Beserra

Alunos :

Bruno Carvalho Faria dos Santos – 14/0132767

Elias Queiroga Vieira – 16/0118719

GRUPO 10

Ponto de controle 4:

Projeto: Music Player

Brasília – Df, 29 de Novembro de 2019.

O pino P1.6 estava entregando tensão abaixo da necessária para ligamento do botão, para isso, o botão foi conectado utilizando a tensão de 5V e com um resistor de $1.2k\Omega$ para fazer um divisor de tensão e entregar a tensão certa.

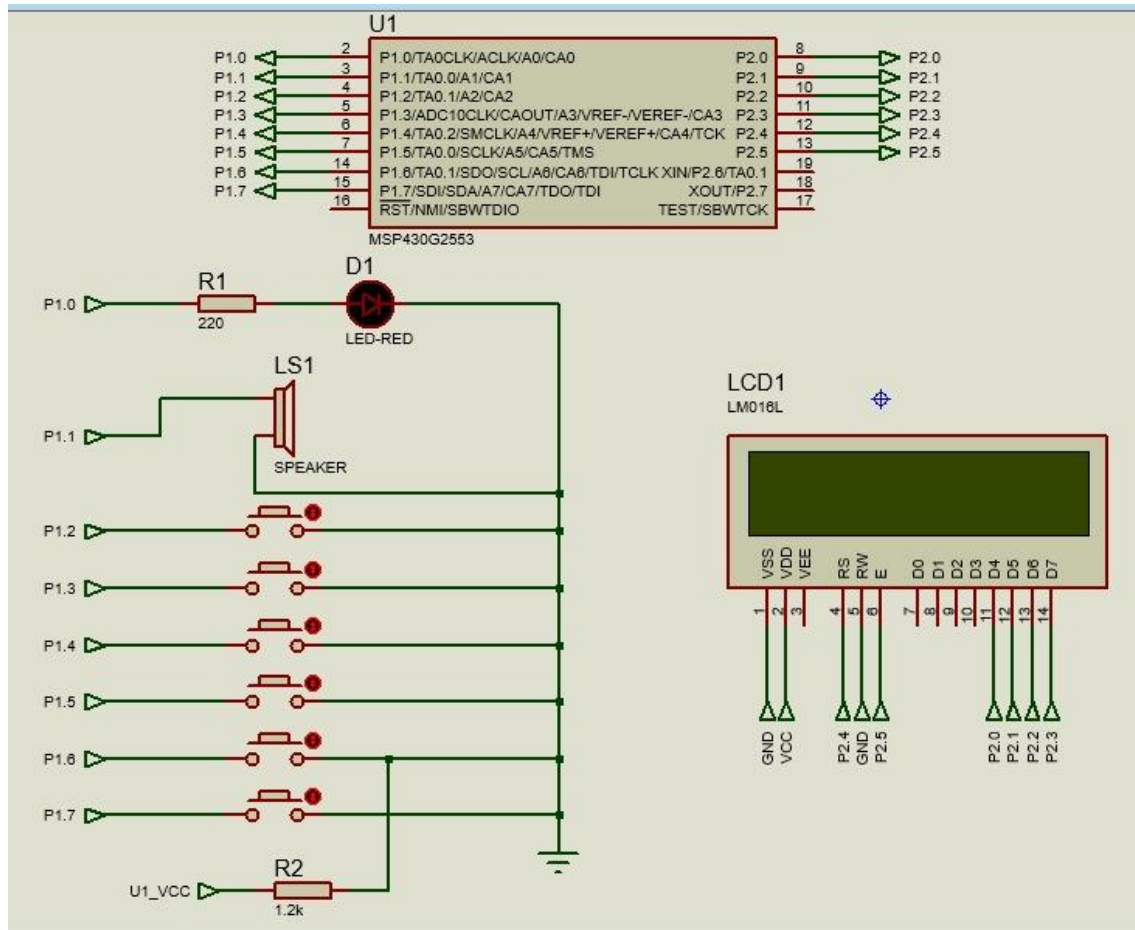


Figura 1. Esquemático da implementação física do projeto.

Todos os requisitos apresentados para o desenvolvimento do projeto foram executados, utilizando os princípios aprendidos em sala uso de WDT, Timer, ISR, comunicação serial com periférico (melhoria proposta no Ponto de controle 2), implementado na placa MSP430G2553, o programa apresentou o resultado esperado.

Como especificado no Ponto de controle 3, em que estava faltante a implementação de alguma rotina em Assembly para o código, era o principal ponto a ser apresentado neste PC4, conseguimos efetuar essa implementação por meio da criação de um arquivo *.asm* que seguia o seguinte formato:

```

        .cdecls C,NOLIST, "msp430.h"    ; Processor specific definitions

;=====
; get_port
;=====

.global  get_port          ; Declare symbol to be exported
        .sect ".text"      ; Code is relocatable
get_port: .asmfunc
        mov.b    &P1IN,R13    ; Store Port 1 into R13 (unused cause
byte)
        and.b    R12,R13      ; use mask
        mov.b    R13,R12      ; Store value into R12 (return
parameter)
        .if ($defined(__MSP430_HAS_MSP430XV2_CPU__) |
$defined(__MSP430_HAS_MSP430X_CPU__))
            reta
        .else
            ret
        .endif
        .endasmfunc

        .end

```

Neste arquivo foi implementado a rotina em assembly, que no caso do nosso projeto é responsável pela configuração dos pinos da porta P1, associados aos BUTTON, LED e SPEAKER, com este arquivo pronto, usando o comando:

```
extern void get_port (void);
```

Responsável por “puxar” o arquivo .asm para o arquivo em C, e dentro da parte da sintaxe do código onde devia ser implementado, era chamado como:

```
get_port();
```