“Homens são de Marte, mulheres são de Vênus e computadores são do inferno.

Lendo entradas e realizando branches

Paulo Ricardo Lisboa de Almeida

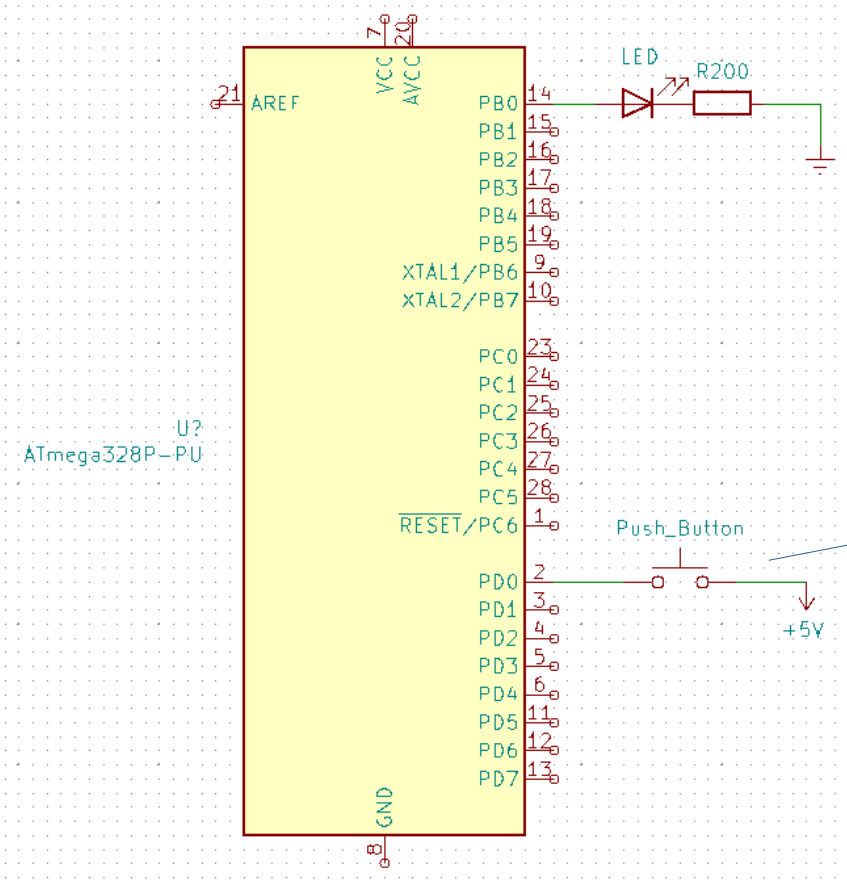
UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

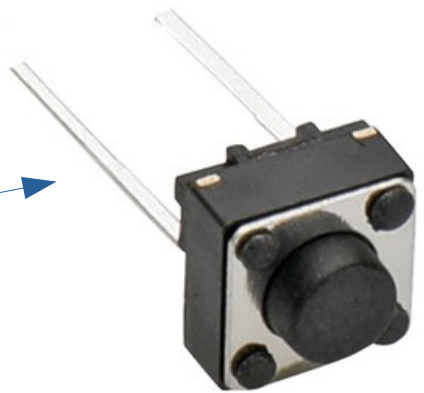
Observações sobre o PIC

- Para transplantar o que será discutido na aula de hoje
  - Aprenda sobre portas de dreno aberto no PIC
    - http://prlalmeida.com.br/microprocessadores2019-02/Aula20.pdf
  - Veja como funciona o I/O no PIC e os branches
    - http://prlalmeida.com.br/microprocessadores2019-02/Aula21.pdf

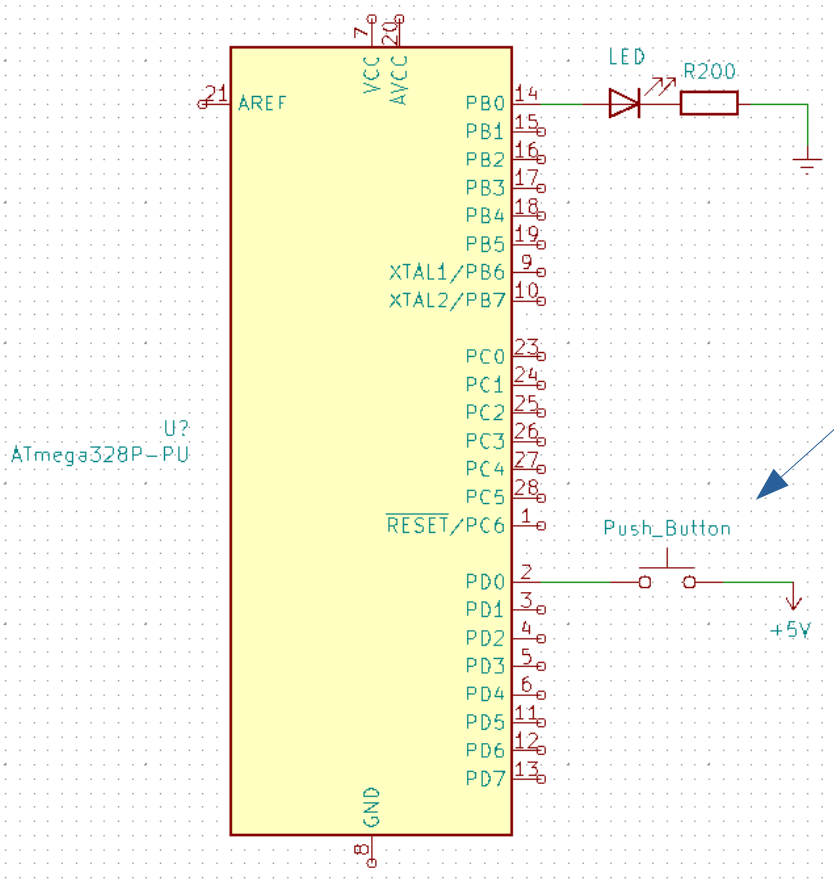
Monte o circuito

Veja a posição dos pinos no manual!!!



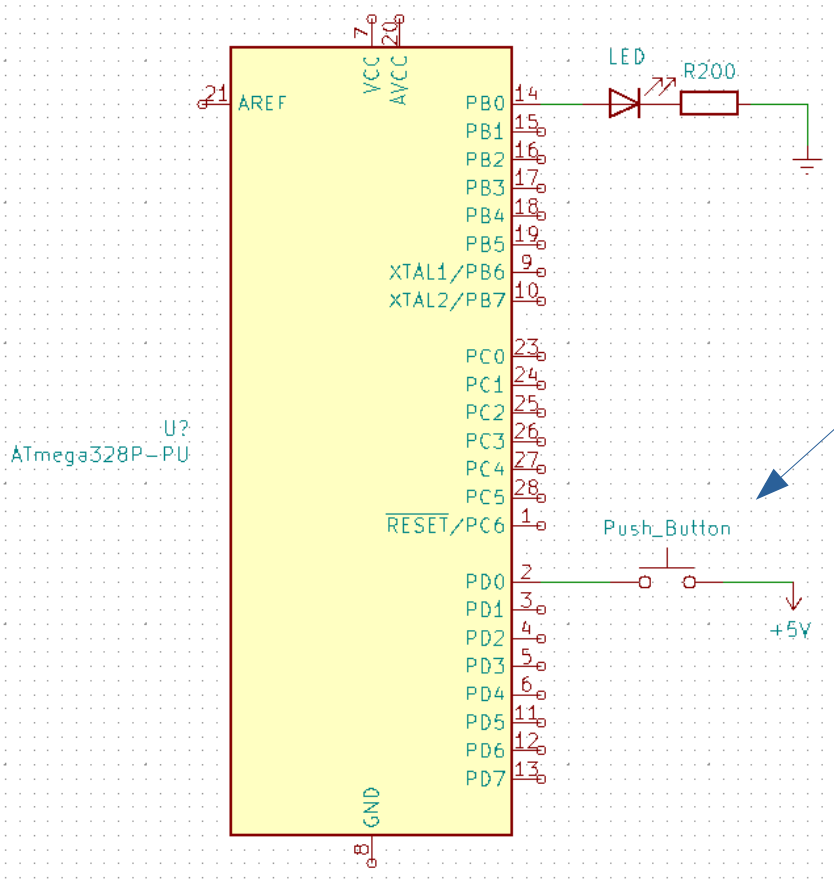


Monte o circuito



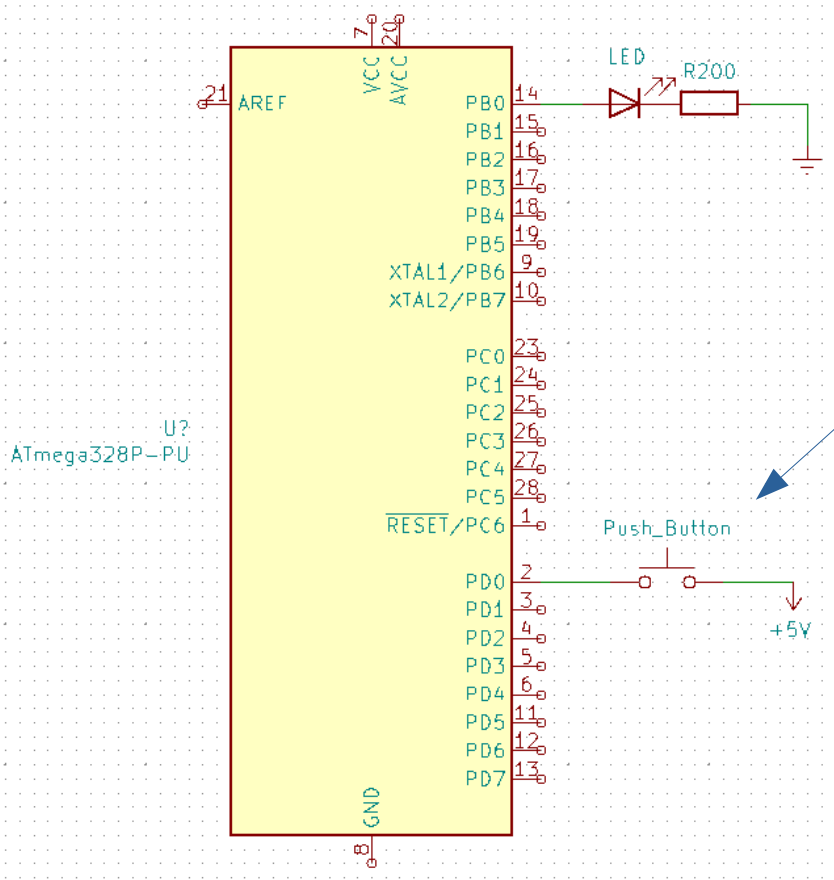
Quando pressionamos o botão, 5V são enviados para PD0 (1 lógico).

Monte o circuito



E quando o botão não está pressionado?

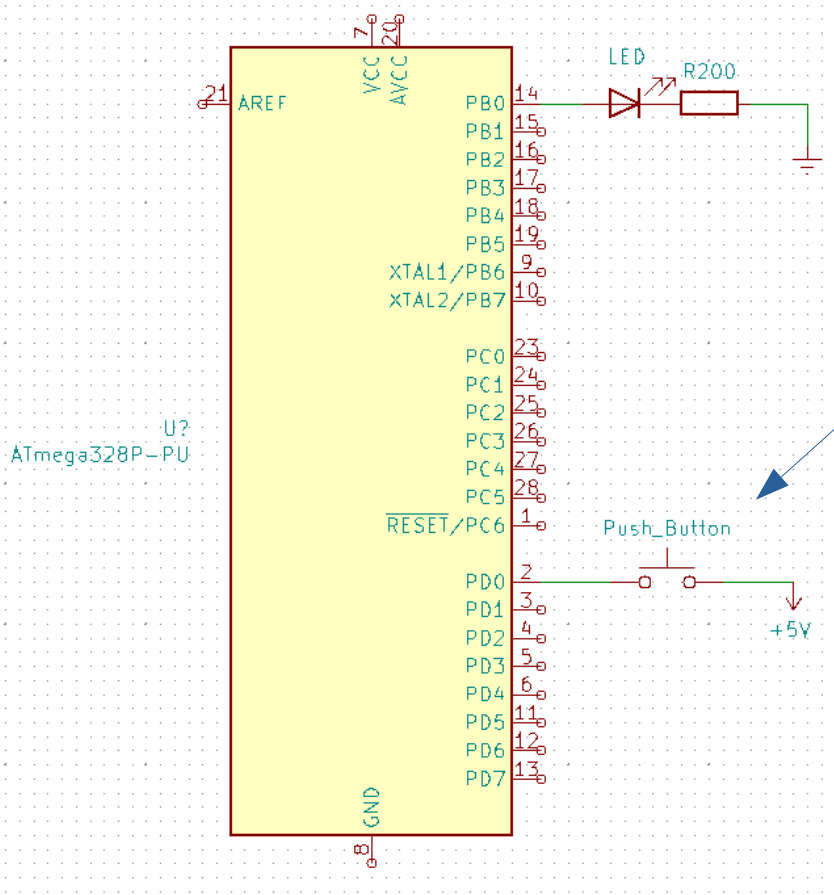
Monte o circuito



Quando o botão não está pressionado, PD0 não está conectado a nada.
Temos um **sinal flutuante**.
Não temos certeza do que está entrando em PD0.
O fio “solto” pode estar operando como uma antena, e captando ruídos externos (e agora, é 0 ou 1???)

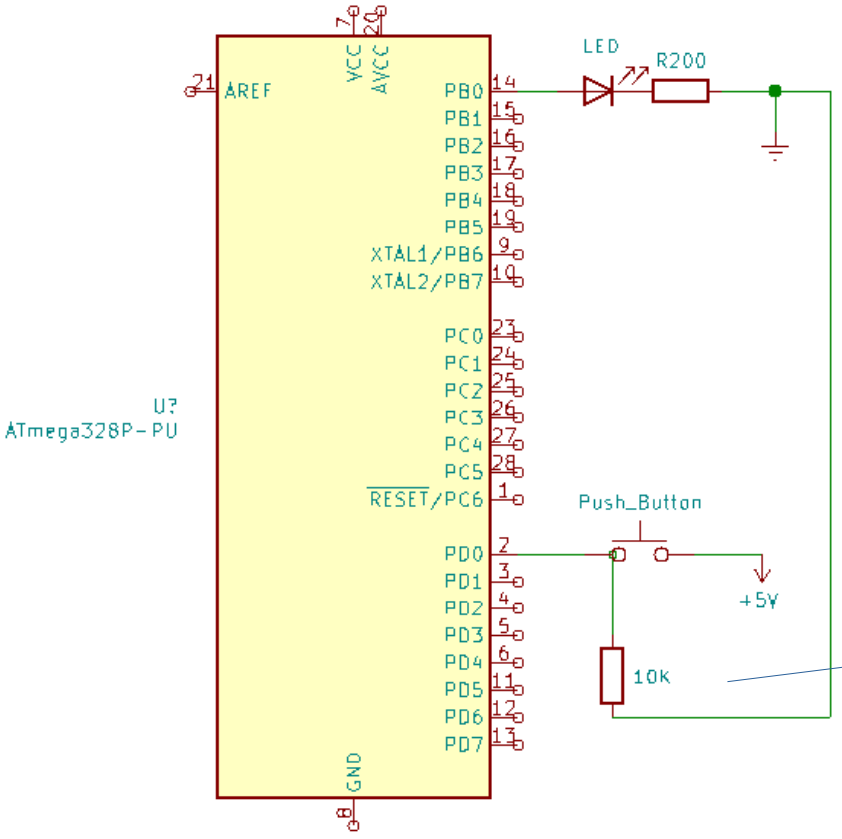
UDESC

Monte o circuito



Como resolver, para que quando o botão não estiver pressionado, 0 lógico seja enviado?

Monte o circuito



Circuito de pull-down. Pesquise!!!

Verifique no Manual

- Como fazer com que o **pino 0 da porta D** seja configurado como **entrada**
  - Quais registradores estão atrelados a Porta D?

Verifique no Manual

- Como fazer com que o **pino 0 da porta D** seja configurado como **entrada**
  - Quais registradores estão atrelados a Porta D?
  - Seção 14 do manual
  - A entrada e saída é **mapeada na memória**
    - DDRD
      - Endereço 0x0A
      - Bits controlam se temos entrada ou saída em cada pino da porta D
    - PORTD
      - Endereço 0x0B
      - Bits controlam o que é **enviado** nos pinos da porta D
    - PIND
      - Endereço 0x09
      - Podemos ler PIND para saber se cada pino individual está lendo 0 ou 1 lógico

Diretiva .equ

- Da mesma forma que no x86-64, podemos definir novos nomes para endereços de memória, registradores, … através de .equ
- Exemplo:

```
.equ DDRB, 0x04
.equ DDRD, 0x0A
.equ PORTB, 0x05
.equ PIND, 0x09
```

Programa

```
#constantes
.equ DDRB, 0x04
.equ DDRD, 0x0A
.equ PORTB, 0x05
.equ PIND, 0x09

    .global main
    .type main, @function
main:
    ldi r16,0b00000001 ;carrega 1 para r16
    out DDRB,r16 ;pino 0 de B como saída
    eor r16, r16 ;carrega 0 para r16
    out DDRD, r16 ;todos pinos de PORTD como entrada
LOOP:
    ldi r17, 0b00000000 ;carrega constante 0
    in  r16, PIND
    sbrc r16,0 ;pule a prox. inst. se o bit 7 está setado
    inc r17 ;some 1 em r17
    out PORTB,r17
    jmp LOOP
```

UDESC

Exercícios

1. Ligue outro LED, agora em PB1. Ao pressionar o botão, o LED em PB0 deve acender, e o em PB1 apagar. Com o botão liberado, PB1 deve acender, e PB0 apagar.

2. Considere a rotina delay_omic disponibilizada. Compreenda como essa rotina funciona. Crie um programa que fica em um loop infinito, onde um LED em PB0 pisca. Utilize a rotina disponibilizada. Pesquise sobre como funcionam as instruções *call* e *ret*.

- Para ambos exercícios, você deve submeter:
  - um vídeo curto do circuíto operando
  - O o código fonte de cada exercício
    - O fonte deve compilar normalmente utilizando o avr-gcc ou o MPLAB (se você estiver usando o PIC)

Referências

- S. Naimi, S. Naimi, M. Mazidi. **The Avr Microcontroller and Embedded Systems Using Assembly and C.** 2010**.**
- **megaAVR® Data Sheet.** Microchip, 2018.
- **ATmega328P Automotive - Complete Datasheet**. Microchip.
- **AVR Instruction Set Manual**. Microchip, 2016.
- D. Patterson; J. Henessy. **Organização e Projeto de Computadores**: **a Interface Hardware/Software.** 5a Edição. Elsevier Brasil, 2017.