

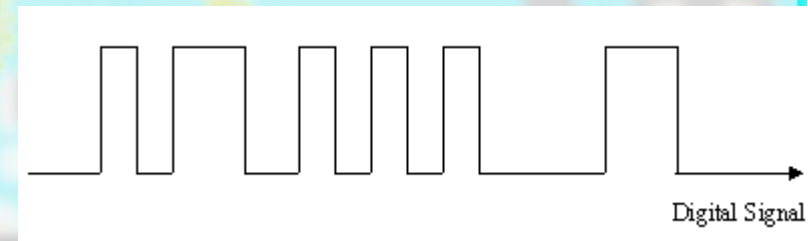
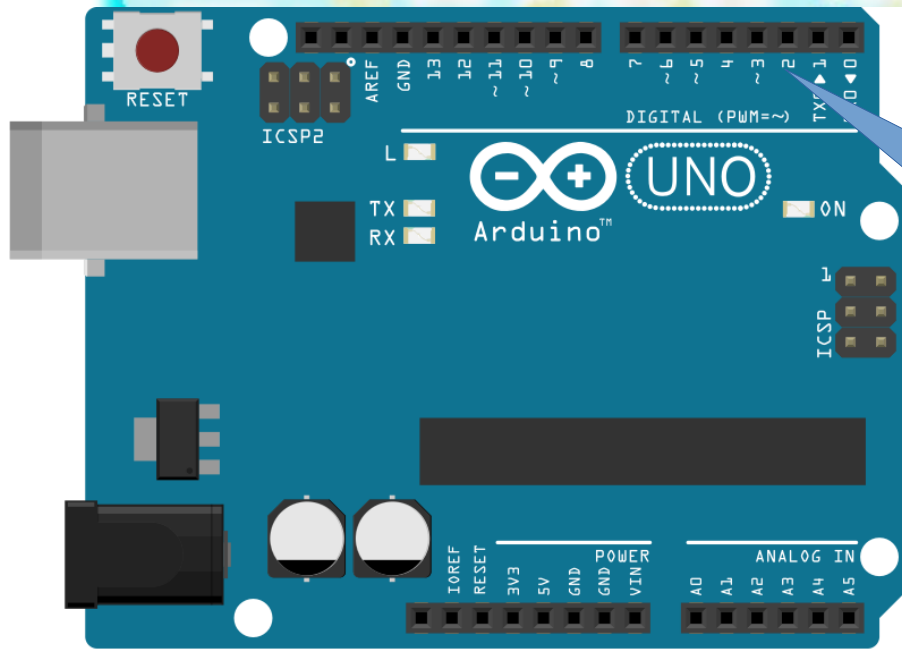
Interrupções externas

Tratamento de callback

Prof. Dr. Roberto Kenji Hiramatsu
Prof. Dr. João Henrique Correia Pimentel

Polling

- Imagine o arduino sentir na porta um sinal “rápido”
- No entanto precisa processar outras coisas



Não é eficiente ficar verificando o que acontece para atender um sinal rápido usando `digitalRead`

Entendendo tratamento de evento externo

```
const byte ledPin = 13;  
const byte interruptPin = 2;  
volatile byte state = LOW;
```

O modificador volatile evita problema de corrompimento da variável acessada por interrupção

```
void setup() {  
  pinMode(ledPin, OUTPUT);  
  pinMode(interruptPin, INPUT_PULLUP);  
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink, CHANGE);  
}
```

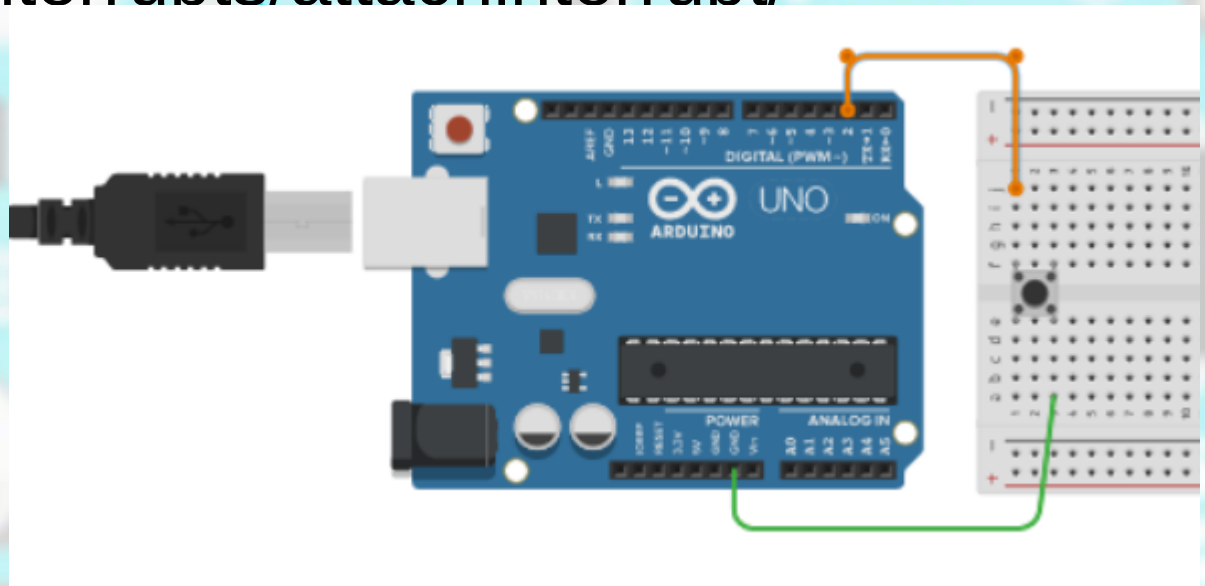
```
void loop() {  
  digitalWrite(ledPin, state);  
}
```

Esta é a rotina de atendimento de interrupção (ISR – Interrupt Service Routine)

```
void blink() {  
  state = !state;  
}
```

Testando um evento externo

- Desenhe o circuito abaixo no tinkercad ou monte no arduino e copie o código encontrado em da página anterior encontrado em:
 - <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>



Pinos que podemos “sentir”

- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);`

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR1000 Rev.1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with CHANGE)



roberto.hiramatsu@ufrpe.br