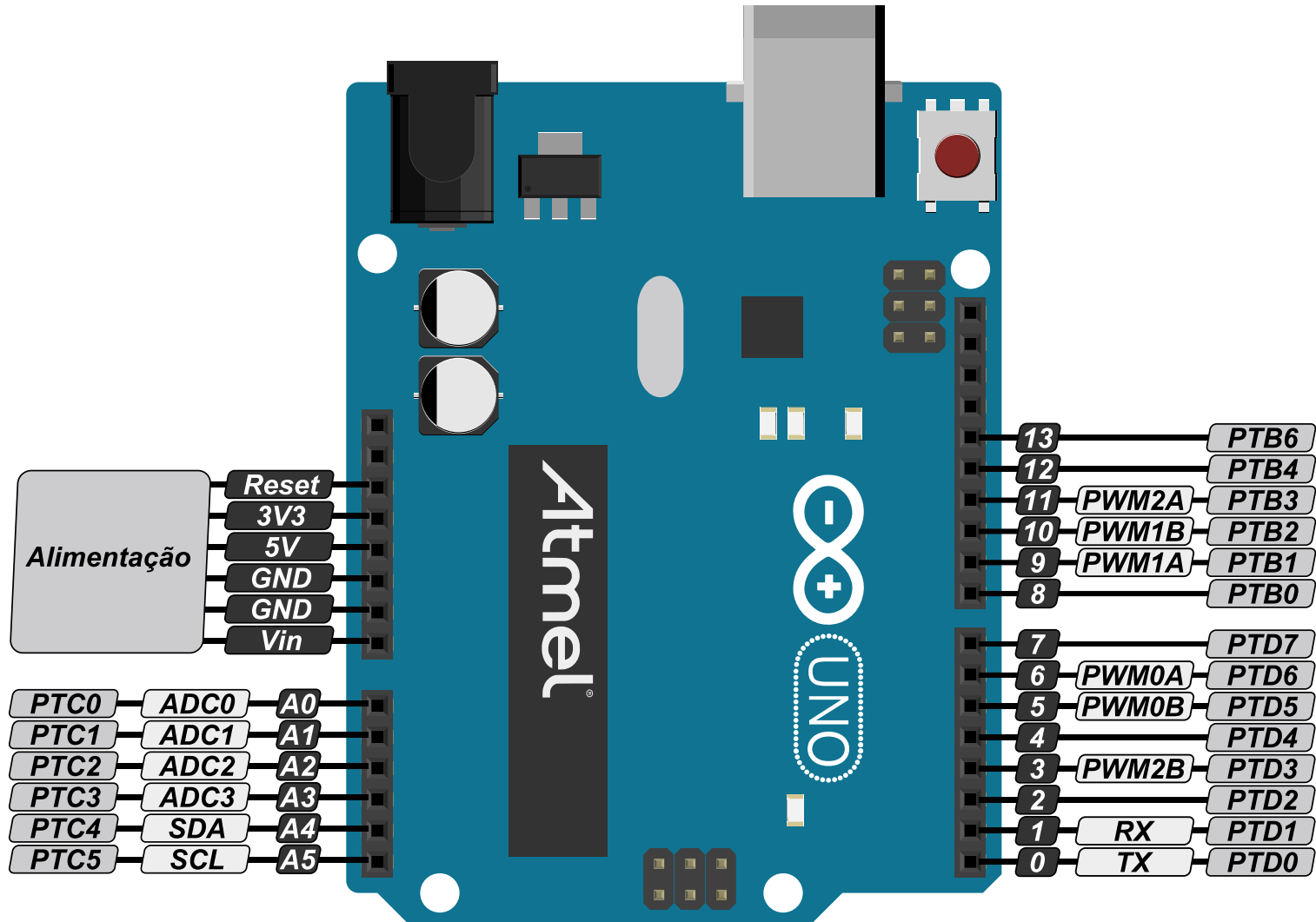


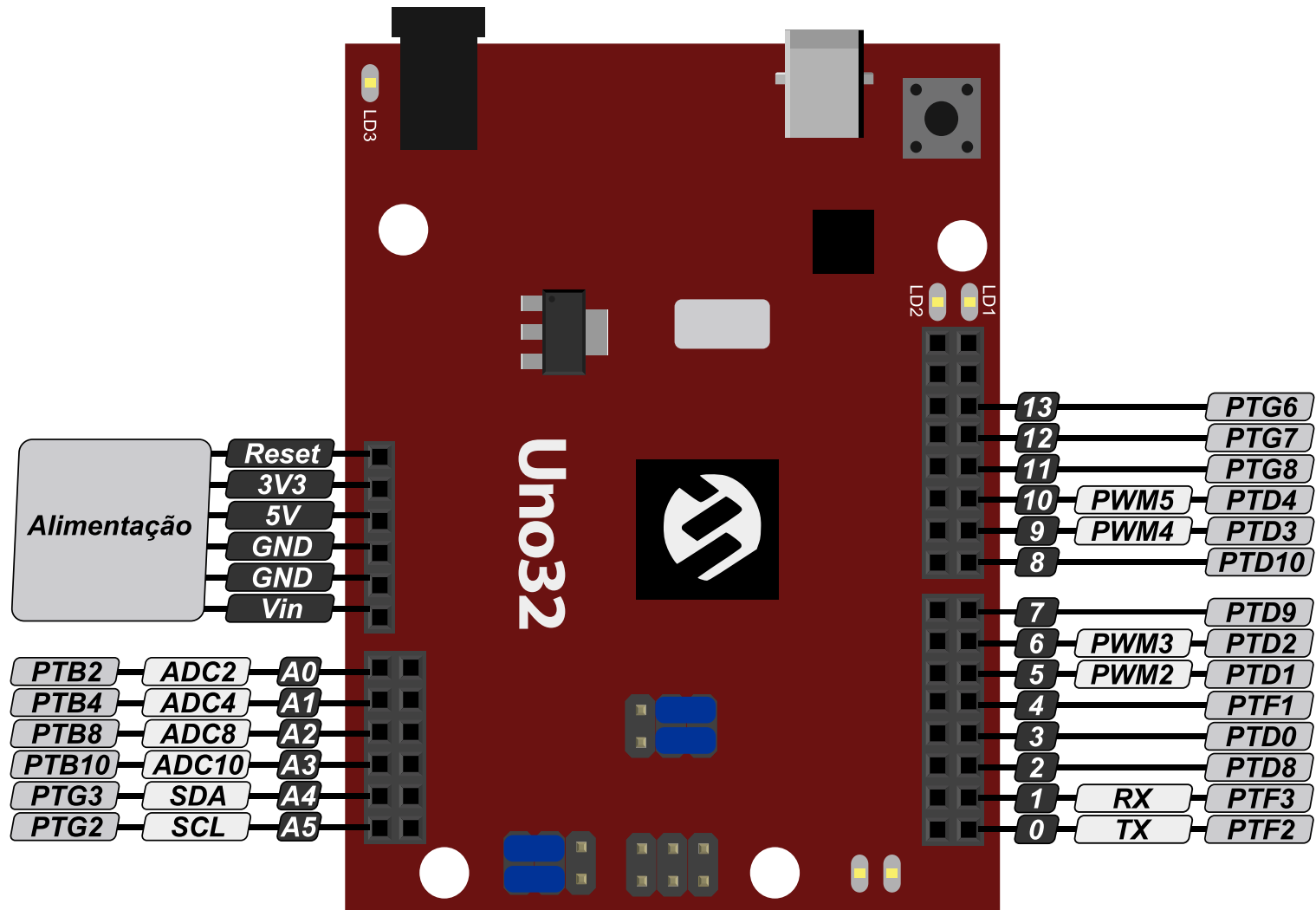
# Prototipação: Interfaces I/O



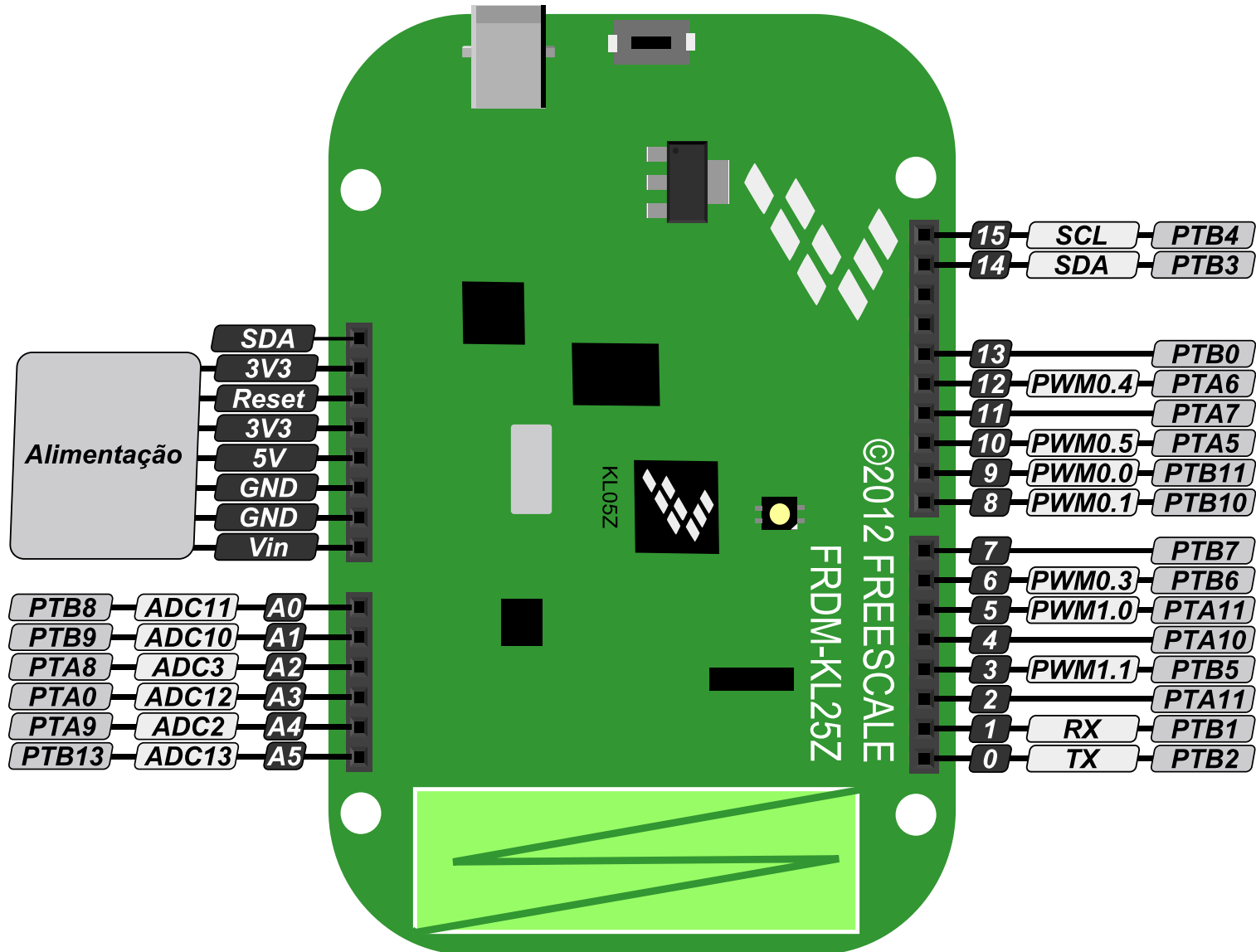
# Conexões dos terminais



# Conexões dos terminais



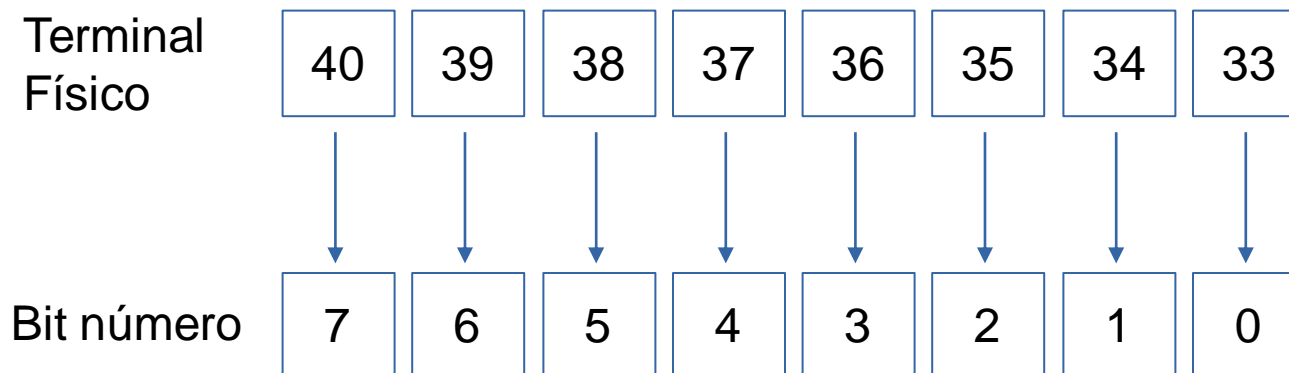
# Conexões dos terminais



Registro	Endereço	Microcontrolador
DDRB	0x04	ATmega328
PORTB	0x05	ATmega328
DDRC	0x07	ATmega328
PORTC	0x08	ATmega328
DDRD	0x0A	ATmega328
PORTD	0x0B	ATmega328
TRISB	0xBF886040	PIC32MX320
PORTB	0xBF886050	PIC32MX320
TRISD	0xBF8860C0	PIC32MX320
PORTD	0xBF8860D0	PIC32MX320
TRISF	0xBF886140	PIC32MX320
PORTF	0xBF886150	PIC32MX320
TRISG	0xBF886180	PIC32MX320
PORTG	0xBF886190	PIC32MX320
PORTA_PDOR	0x4004F000	KL05z32
PORTA_PDIR	0x4004F010	KL05z32
PORTA_PDDR	0x4004F014	KL05z32
PORTB_PDOR	0x4004F040	KL05z32
PORTB_PDIR	0x4004F050	KL05z32
PORTB_PDDR	0x4004F054	KL05z32

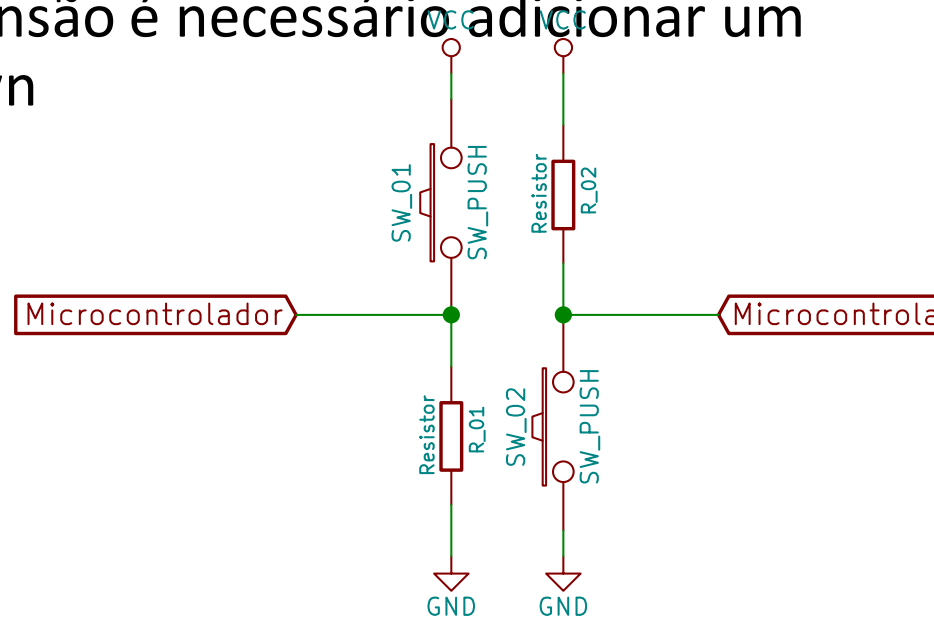
# Periféricos de entrada/saída

- As portas são registros que repassam a informação da memória para os terminais físicos.
  - Cada um dos bits representa um terminal.
  - Ligar o bit -> terminal com tensão alta (3v3 ou 5v).
  - Desligar o bit -> o terminal com tensão zero.
- Por exemplo
  - os oito bits (0 à 7) estão diretamente conectados aos terminais físicos de número 33 à 40.



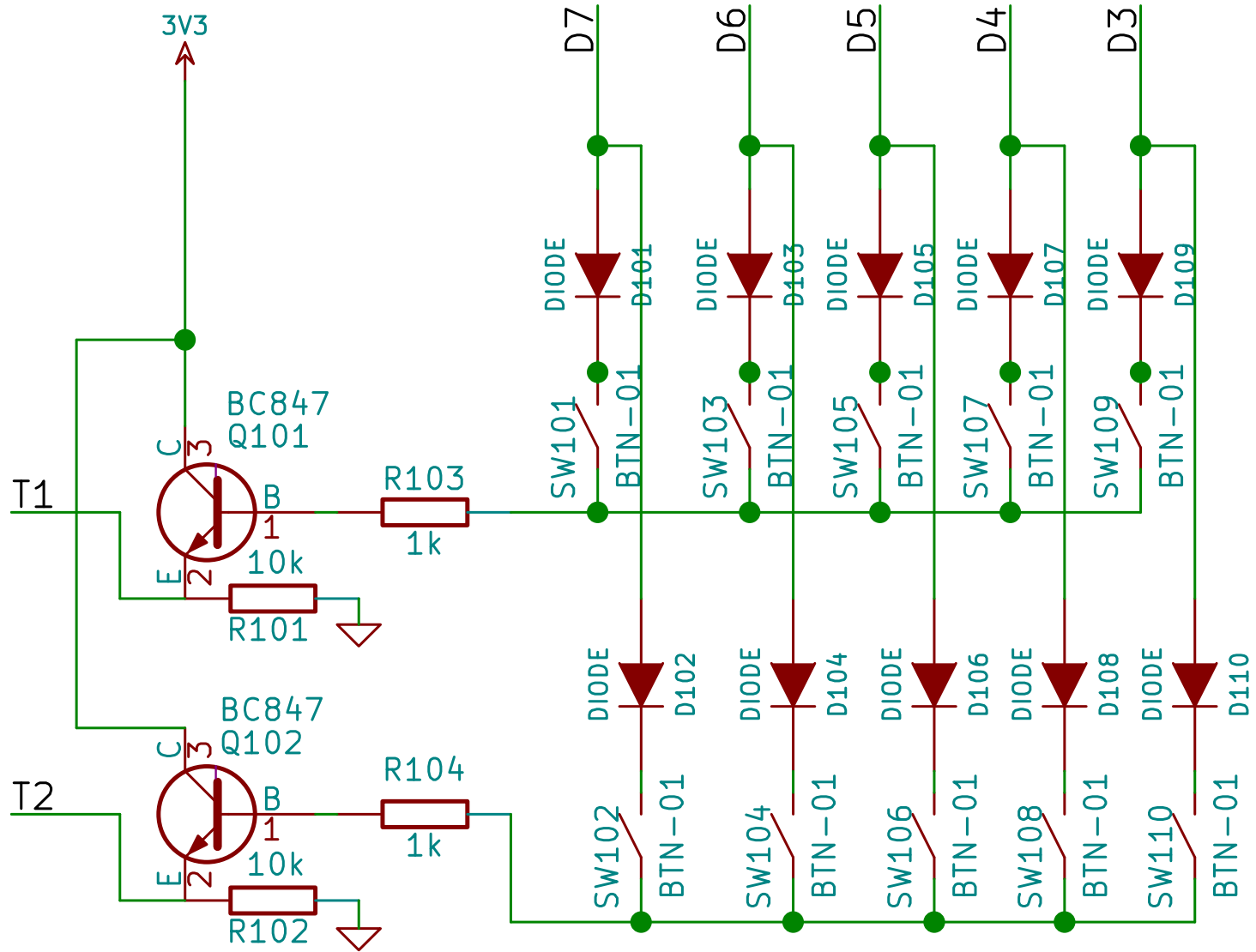
# Entrada digitais

- Criação de um circuito para entrada de informação digital
  - A saída deve possuir apenas dois estados
  - O consumo de energia deve ser o mínimo possível
  - As tensões devem ser compatíveis com o circuito (TTL, CMOS, etc...)
- O circuito mais comum são os botões/teclas
  - A maioria dos botões são possuem dois contatos, para formar os dois níveis de tensão é necessário adicionar um resistor: pullup ou pulldown





# Leitura matricial



# Leitura matricial

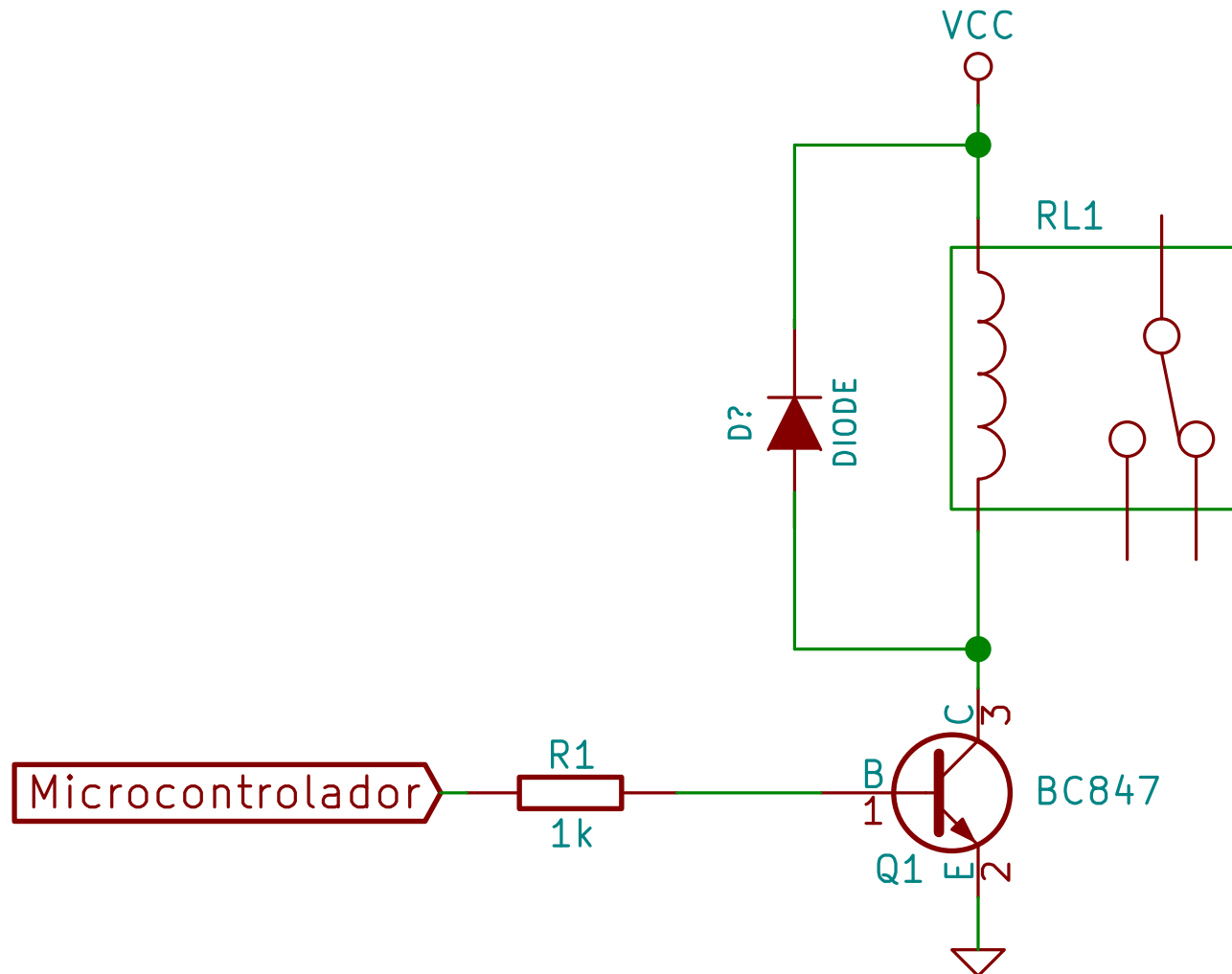
- Na leitura por varredura matricial temos:
  - Aumento da quantidade de chaves que podem ser lidas,
    - Para N terminais disponíveis no microcontrolador, temos um aumento de N chaves para  $N^2$  chaves
  - Aumento da complexidade do software
  - Atraso na detecção de eventos (devido à varredura)
- O processo pode ser descrito como:
  1. Desligar todas as colunas
  2. Ligar apenas a coluna X
  3. Aguardar um tempo para estabilização dos sinais
  4. Realizar leitura nos terminais de entrada
  5. Passar para a próxima coluna  $X = X + 1$

# Outras saídas digitais

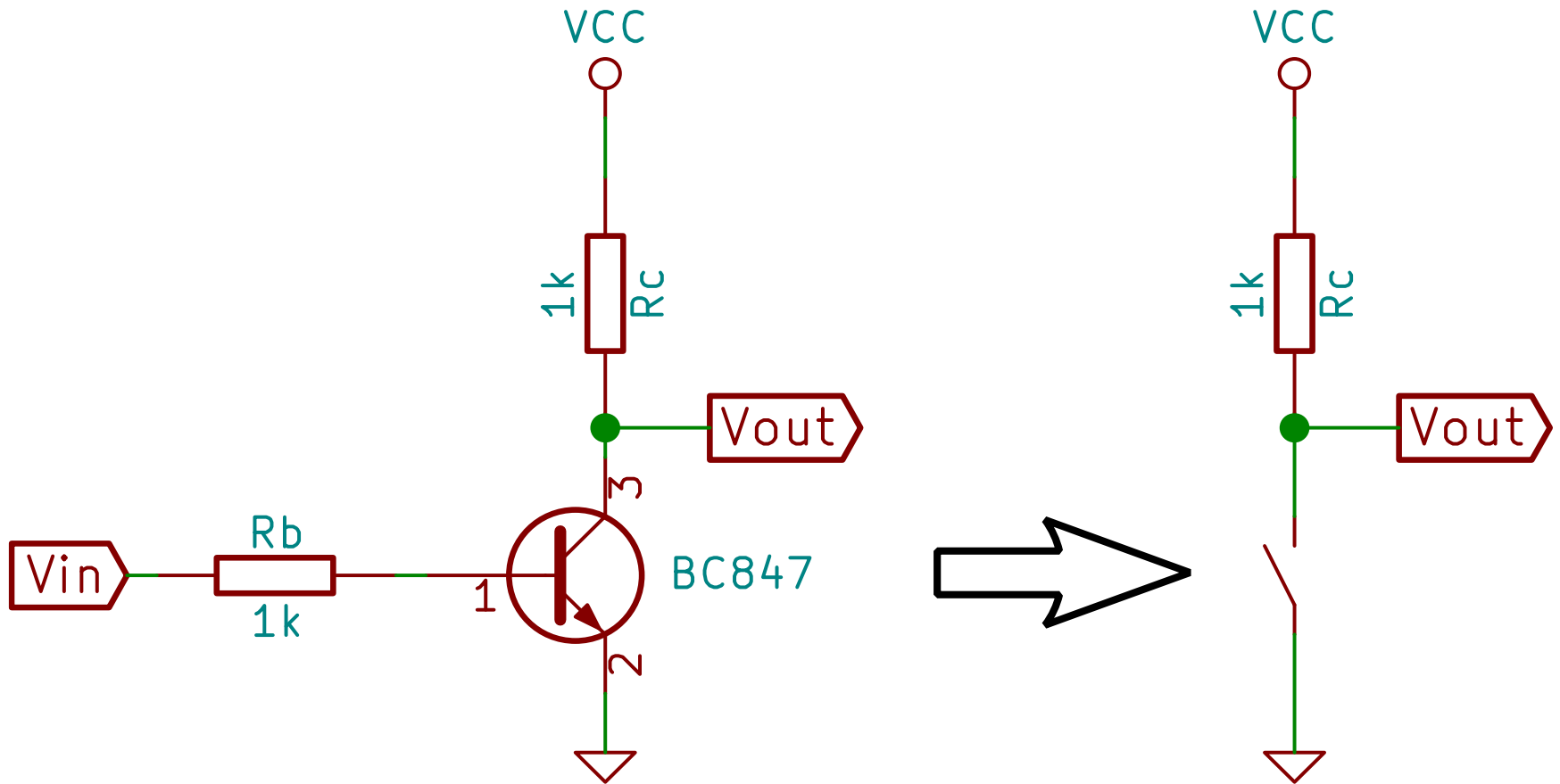
- Transistores
  - Pode funcionar como um amplificador ou chave.
  - No modo amplificador ele possui a capacidade de ampliar o nível de tensão.
  - Como chave ele permite ligar cargas em sua saída.
  - Apenas tensão contínua
- Relés
  - Similar ao transistor operando como chave, mas permite DC e AC
- Relés de estado sólido
  - Circuitos eletrônicos que funcionam como um relé.
  - Dependendo do modelo pode permitir AC



# Acionamento por relé

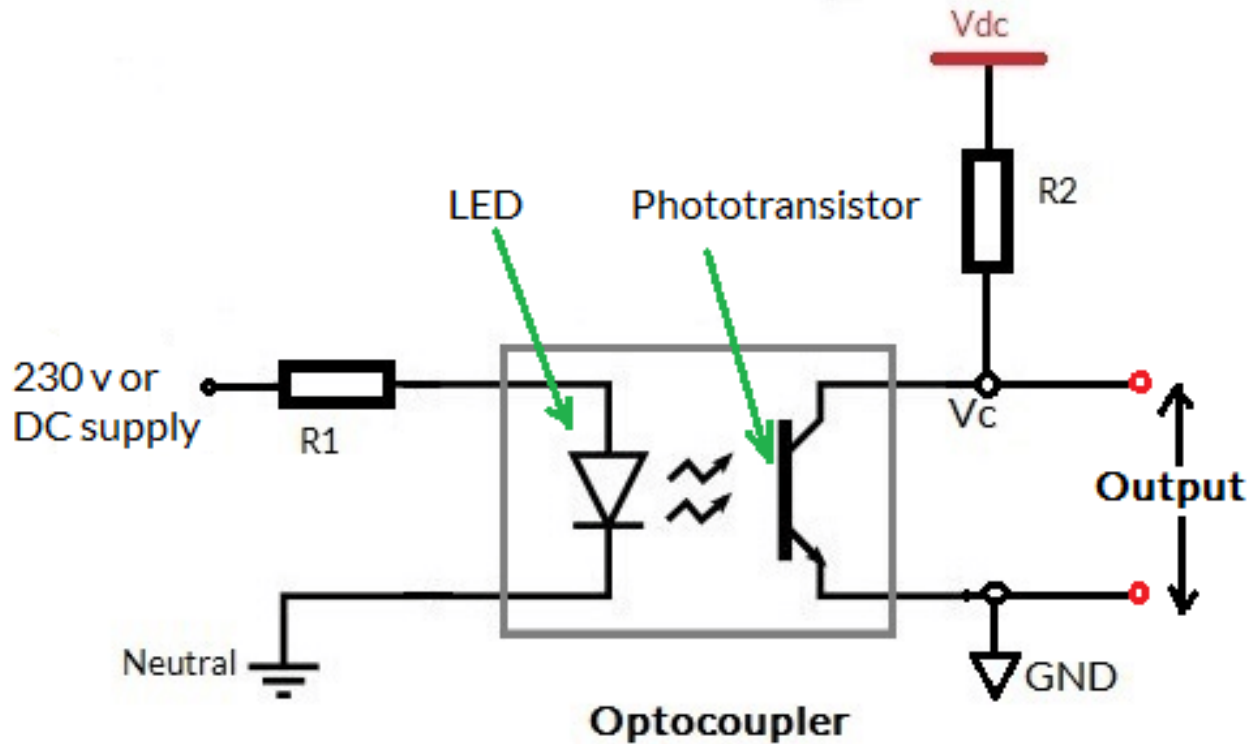


# Transistor operando como chave

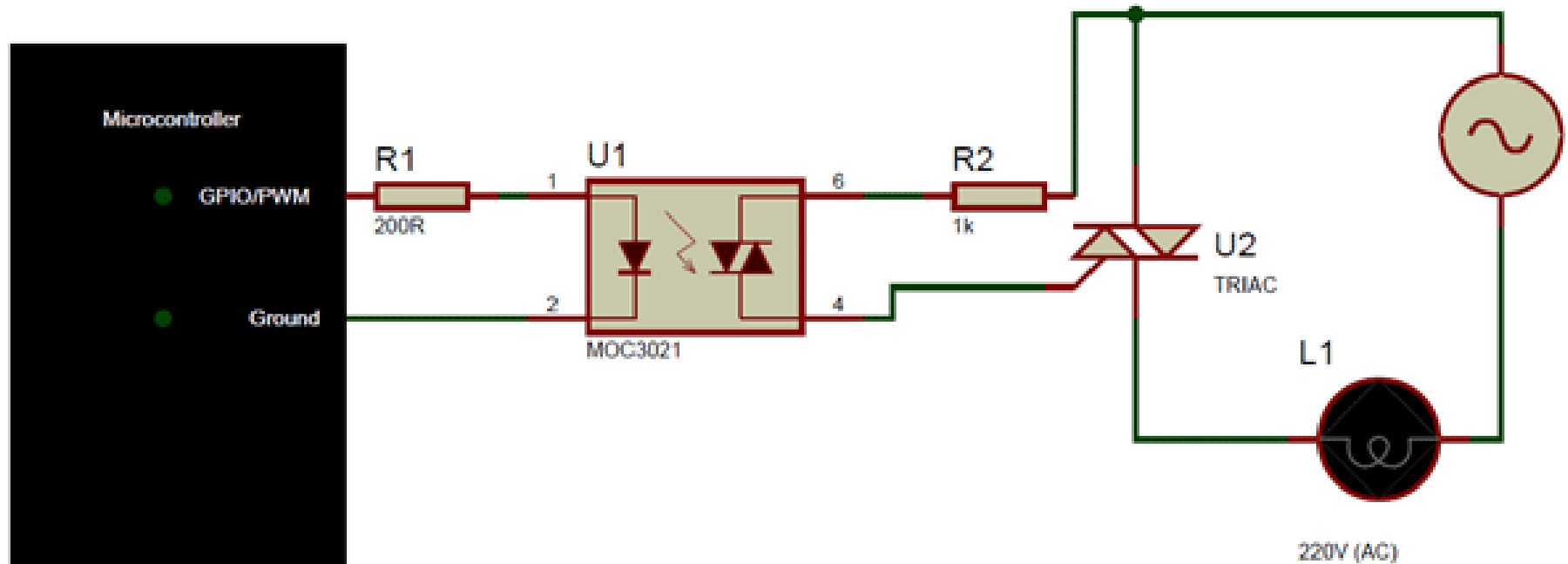


# Entrada digitais

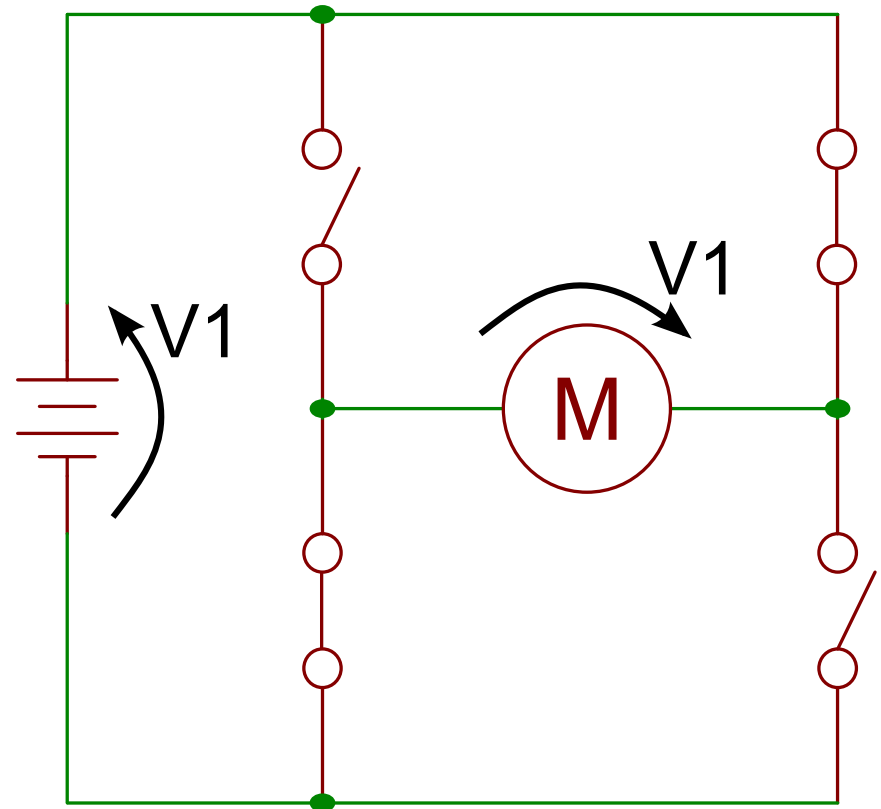
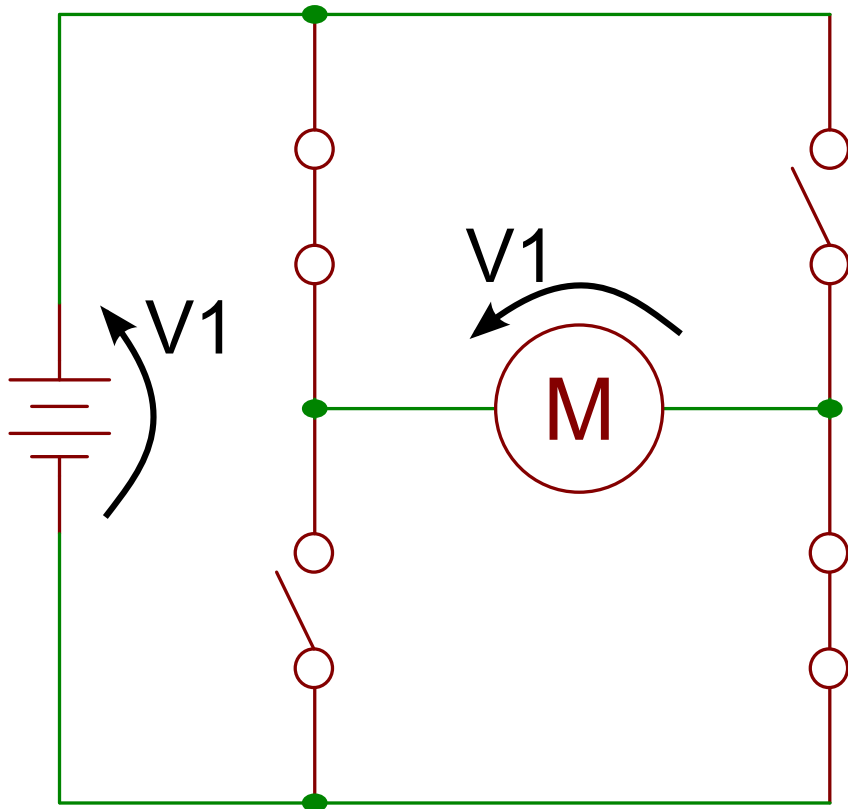
- Entrada AC



# Saídas AC



# Exemplo de uso ponte H

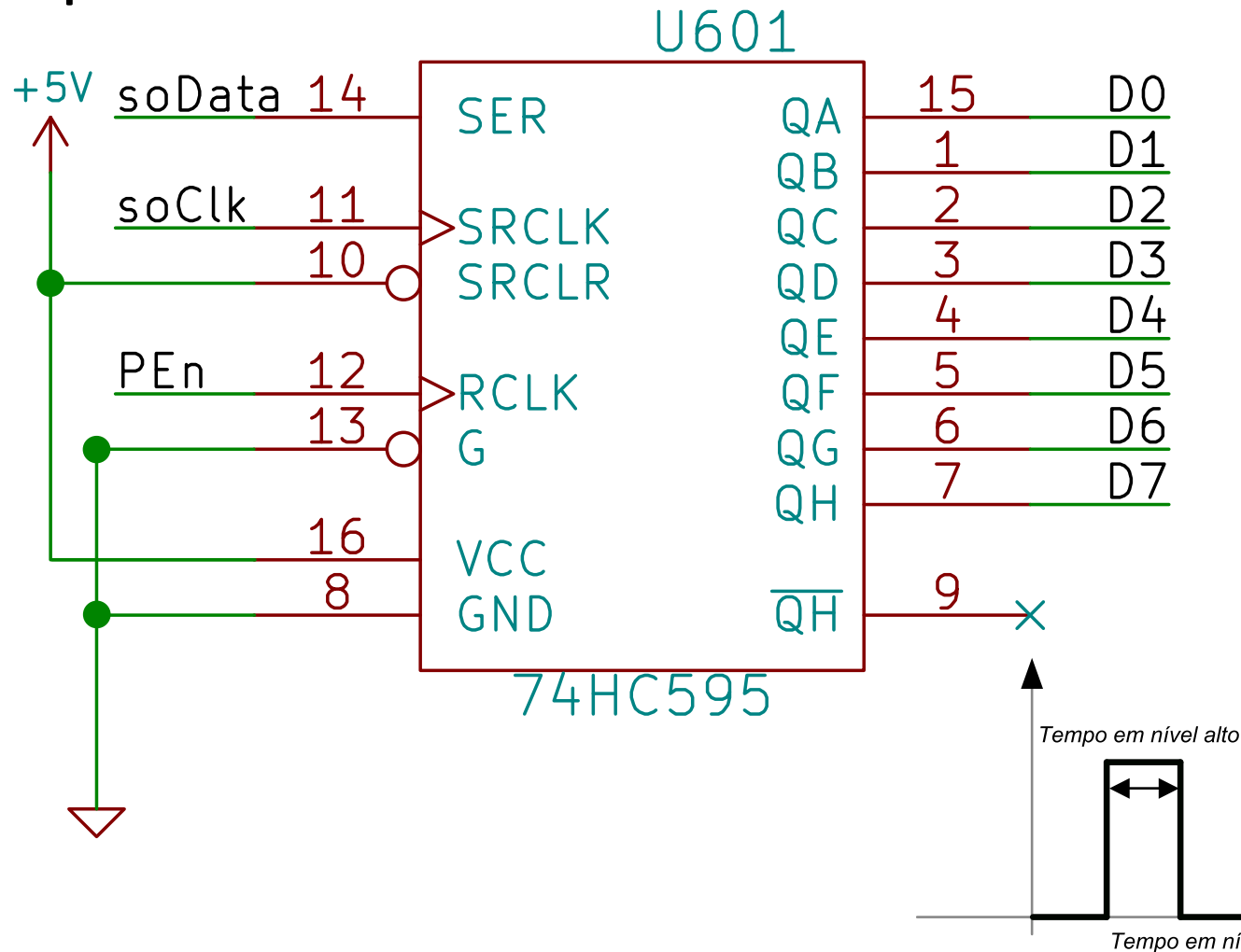




# Expansão de saídas

- O custo de um microcontrolador é dependente da quantidade de saídas disponíveis.
- Se o micro não possui saídas suficientes podemos utilizar circuitos para expansão de saídas
  - conversores de serial para paralelo
  - expansores de IO
  - multiplexação temporal dos terminais
  - multiplexação em frequências diferentes.
- Toda abordagem insere algum problema
  - custo no sistema;
  - atraso na resposta;
  - aumento da complexidade.

# Expansão de saídas



Cada bit é enviado de modo serial pelo terminal soData, o fim da transmissão é enviado um pulso em soClk

# Comunicação Serial

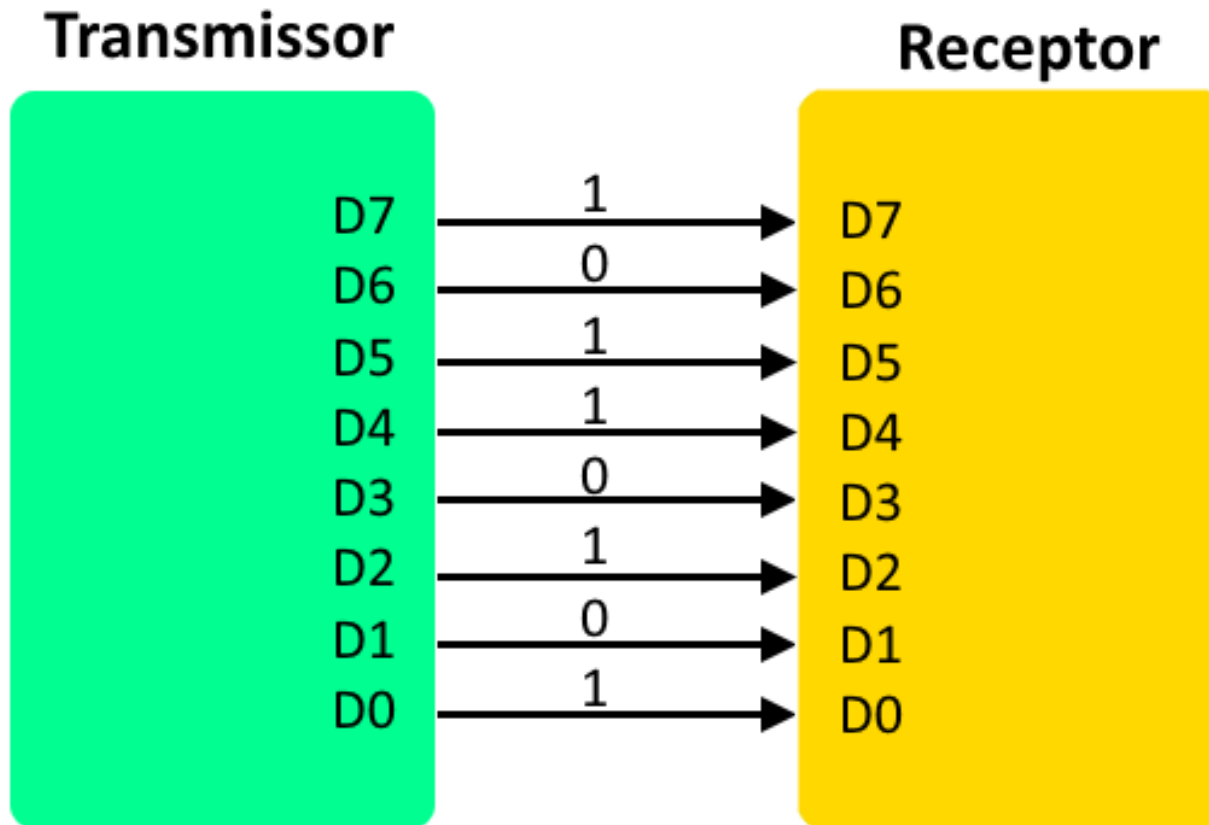
## USART, I2C e SPI

# Padrões de comunicação

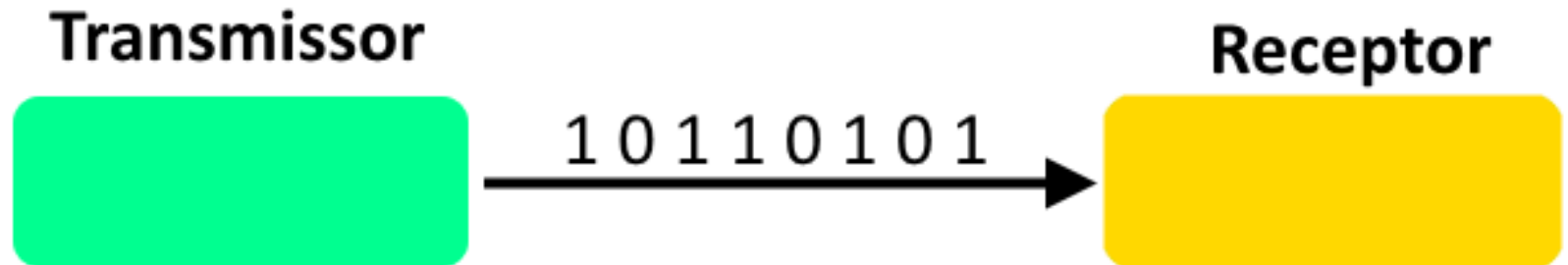
Protocolo	Taxa (bits/s)	Taxa (bytes/s)
Serial MIDI	31.25 kbit/s	3.9 kB/s
Serial EIA-232 max.	230.4 kbit/s	28.8 kB/s
Serial UART max	2.7648 Mbit/s	345.6 kB/s
I2C	3.4 Mbit/s	425 kB/s
Serial EIA-422 max.	10 Mbit/s	1.25 MB/s
SPI Bus (Up to 100MHz)	100 Mbit/s	12.5 MB/s
USB super speed (USB 3.0)	5 Gbit/s	625 MB/s
HDMI v. 1.3	10.2 Gbit/s	1.275 GB/s
Ultra DMA ATA 133 (paralelo 16 bits)	1,064 Mbit/s	133 MB/s
Serial ATA 3 (SATA-600)	6,000 Mbit/s	600 MB/s
Ultra-640 SCSI (paralelo 16 bits)	5,120 Mbit/s	640 MB/s
Serial Attached SCSI (SAS) 3	9,600 Mbit/s	1,200 MB/s

(PCINT14/RESET)	PC6	Pin1	1	28	Pin28 PCS (ADCS/SCL/PCINT13)
➡ (PCINT16/RXD)	PD0	Pin2	2	27	Pin27 PD4 (ADC4/SDA/PCINT12)
➡ (PCINT17/TXD)	PD1	Pin3	3	26	Pin26 PD3 (ADC3/PCINT11)
(PCINT18/INT0)	PD2	Pin4	4	25	Pin25 PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1)	PD3	Pin5	5	24	Pin24 PC1 (ADC1/PCINT9)
	PD4	Pin6	6	23	Pin23 PC0 (ADC0/PCINT8)
	Vcc	Pin7	7	22	Pin22 GND
	GND	Pin8	8	21	Pin21 AREF
(PCINT6/XTAL1/TOSC1)	PB6	Pin9	9	20	Pin20 AVCC
(PCINT7/XTAL2/TOSC2)	PB7	Pin10	10	19	Pin19 PBS (SCK/PCINT5)
(PCINT21/OC0B/T1)	PD5	Pin11	11	18	Pin18 PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0)	PD6	Pin12	12	17	Pin17 PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1)	PD7	Pin13	13	16	Pin16 PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1)	PB0	Pin14	14	15	Pin15 PB1 (OC1A/PCINT1)

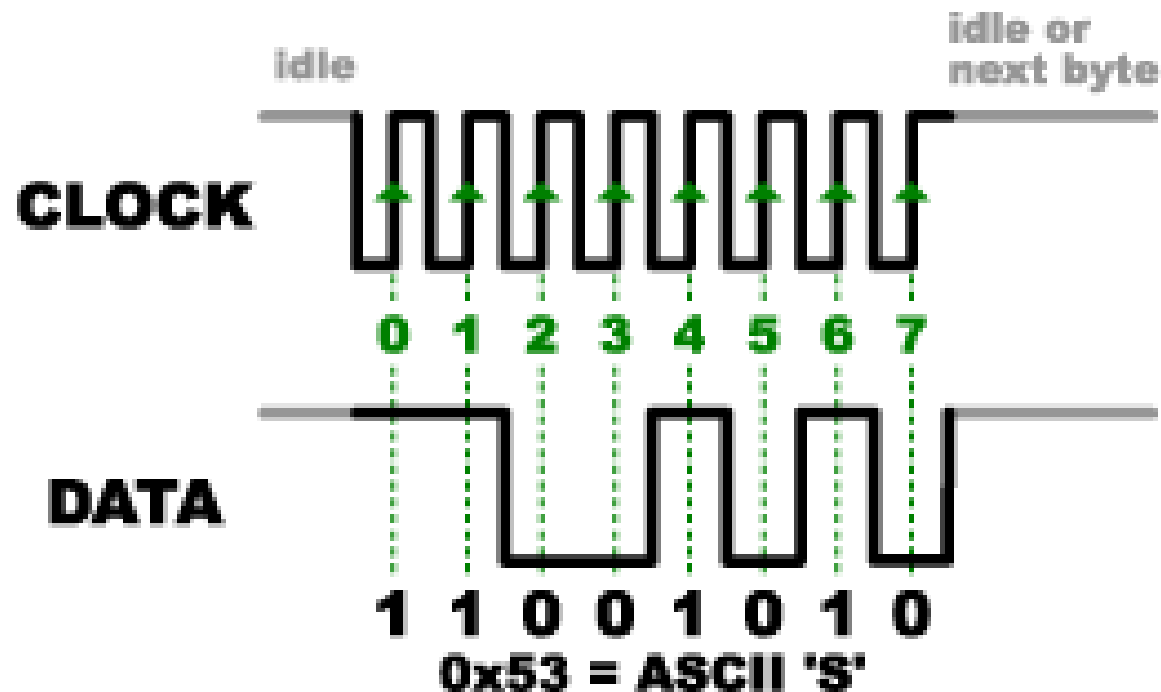
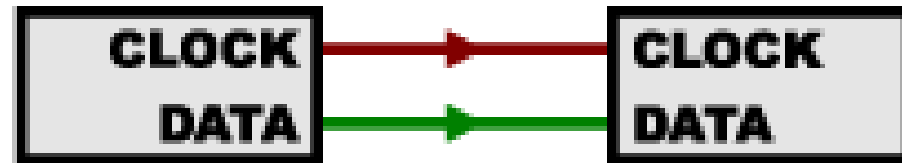
# Comunicação Paralela



# Comunicação Serial

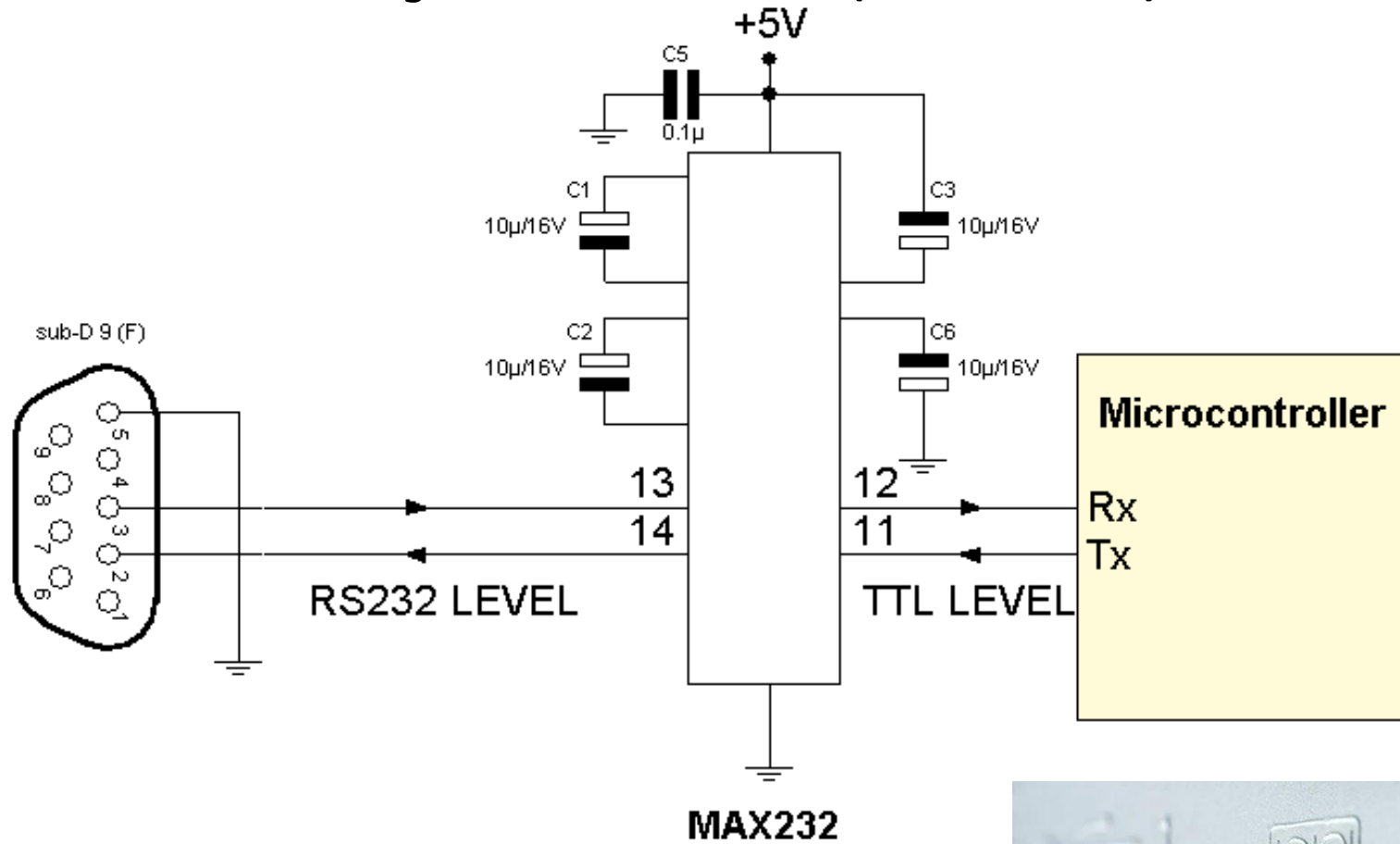


# Comunicação Serial Síncrona e Assíncrona



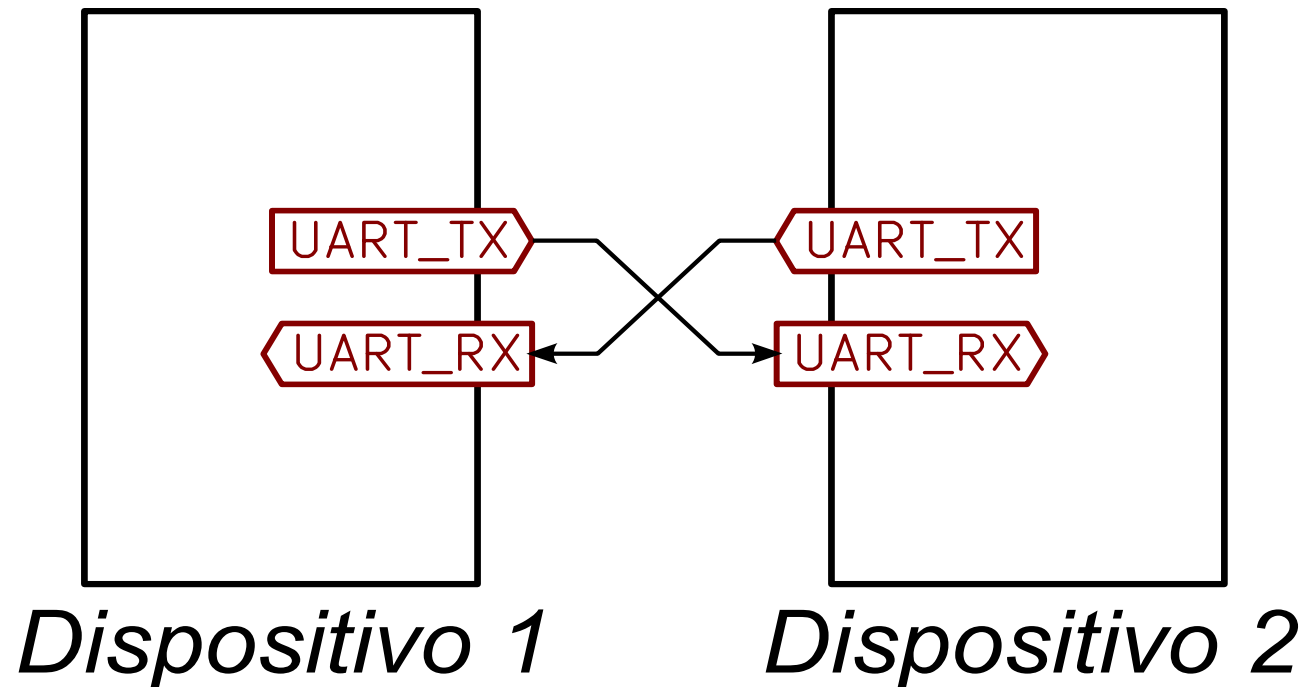


# Comunicação Serial (RS232)



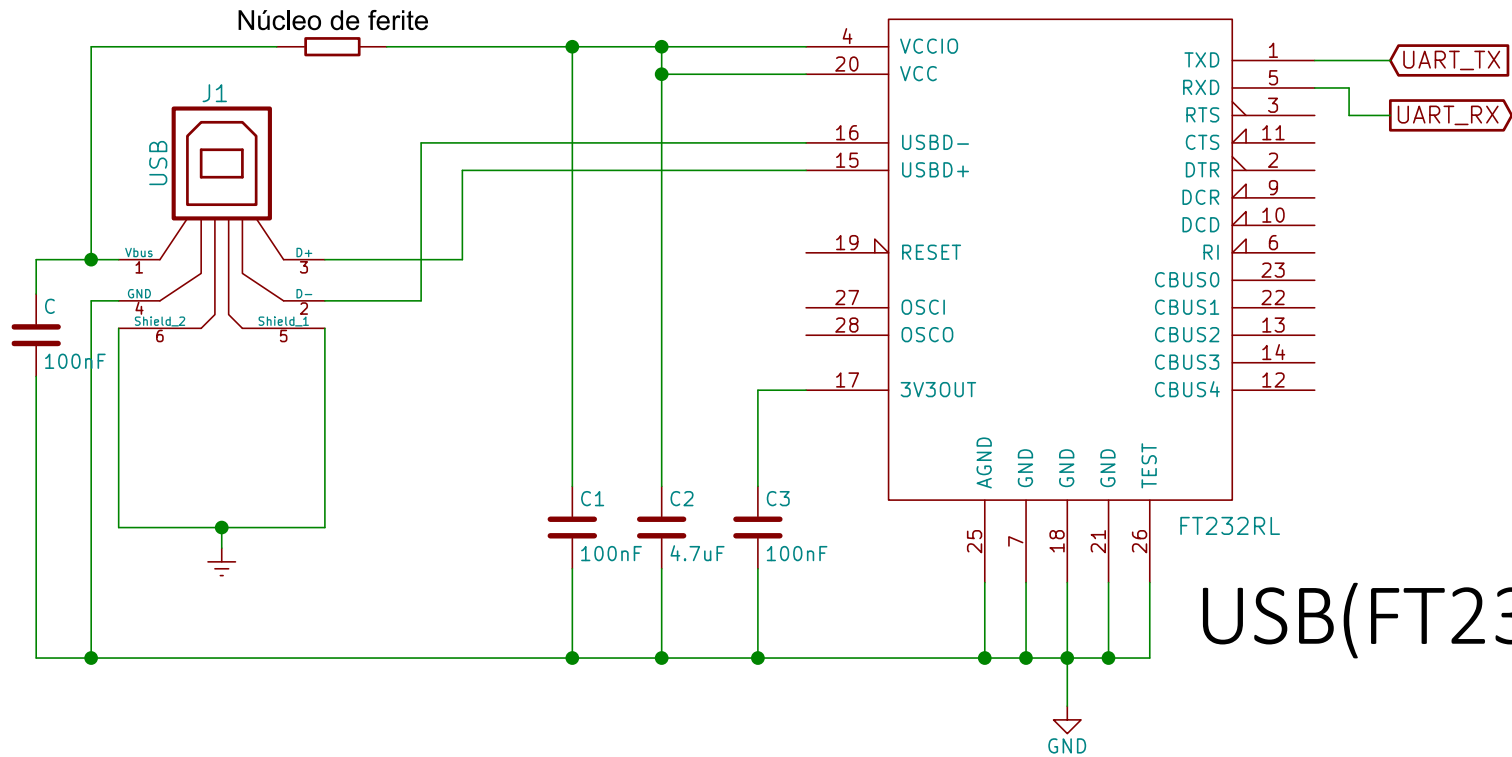
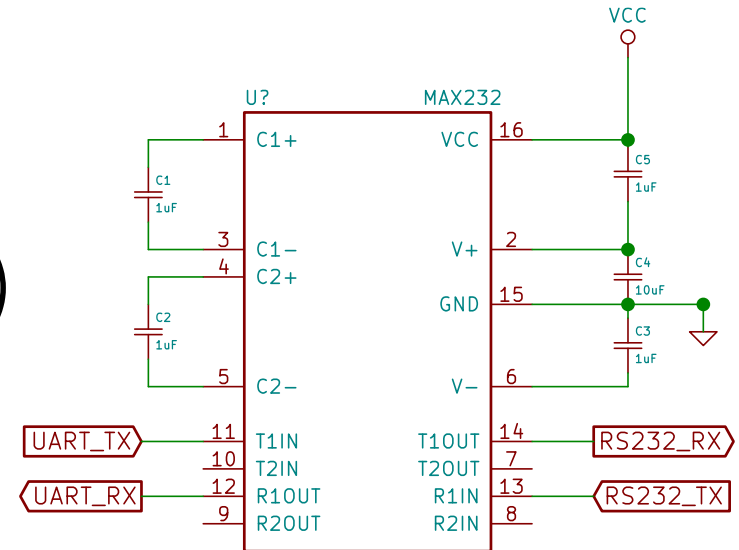
# RS232 versus UART

- Similaridades, diferenças e observações:
  - UART é uma versão generalizada dos protocolos 232, 485, 422, EIA.
  - Visa fornecer uma base padrão para todos estes protocolos
  - As tensões são em geral no mesmo nível do processador
- As conversões para os protocolos ficam a cargo de chips externos
  - MAX232 (ICL232, ST232, ADM232, HIN232)
  - FT232
  - SP3485



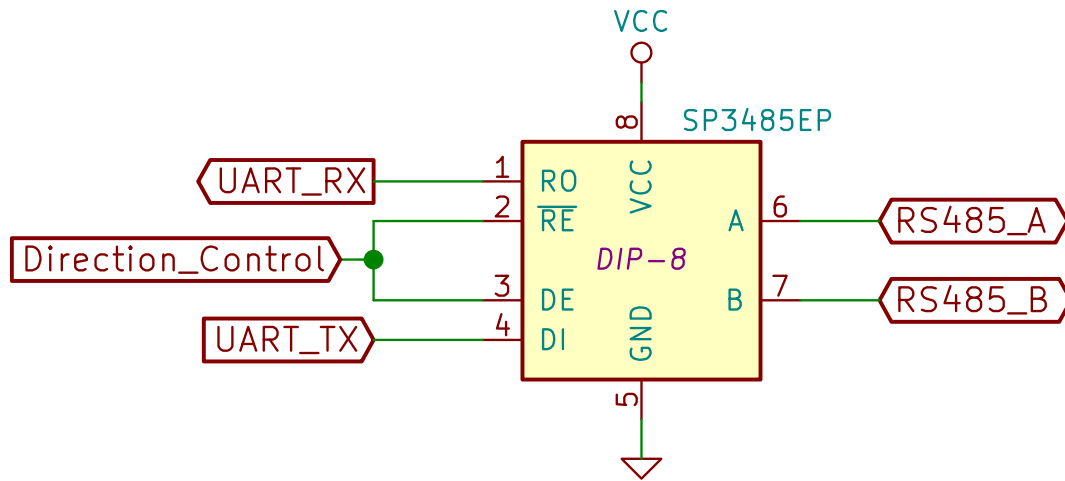
# Transceptores Seriais

## Uart(MAX232)



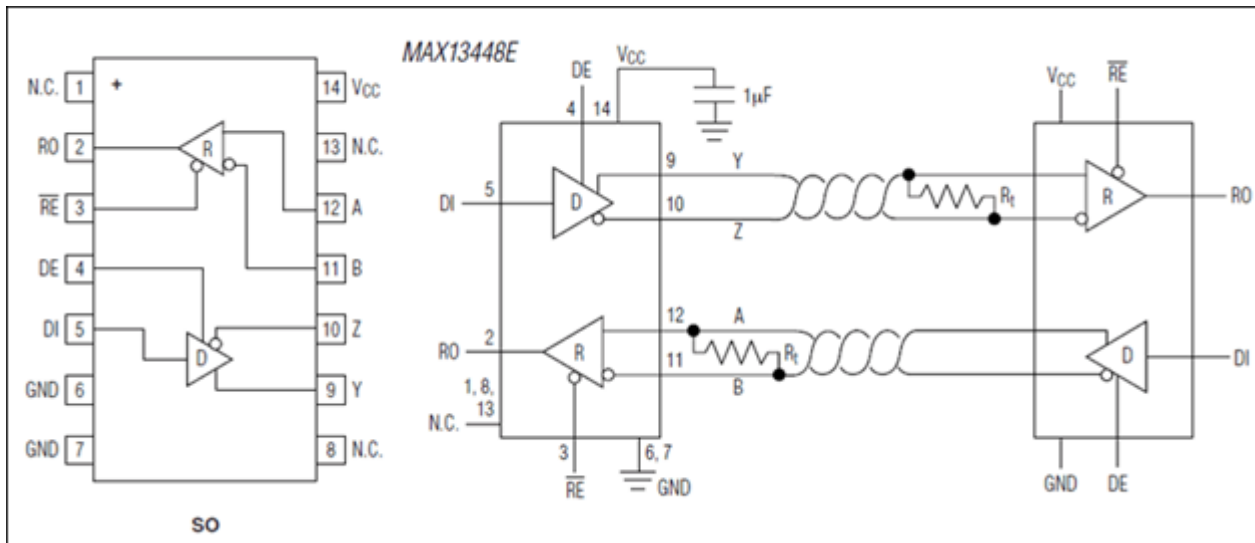
## USB(FT232RL)

# Transceptores Seriais

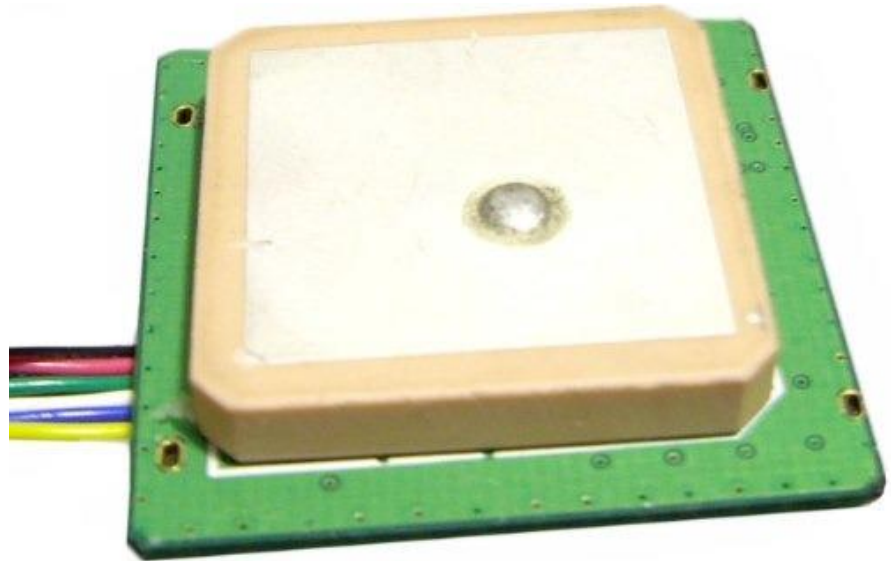


RS485(MAX485)  
Half-Duplex)

RS422(MAX1344)  
Full-Duplex)



# Exemplo de interface com dispositivo UART



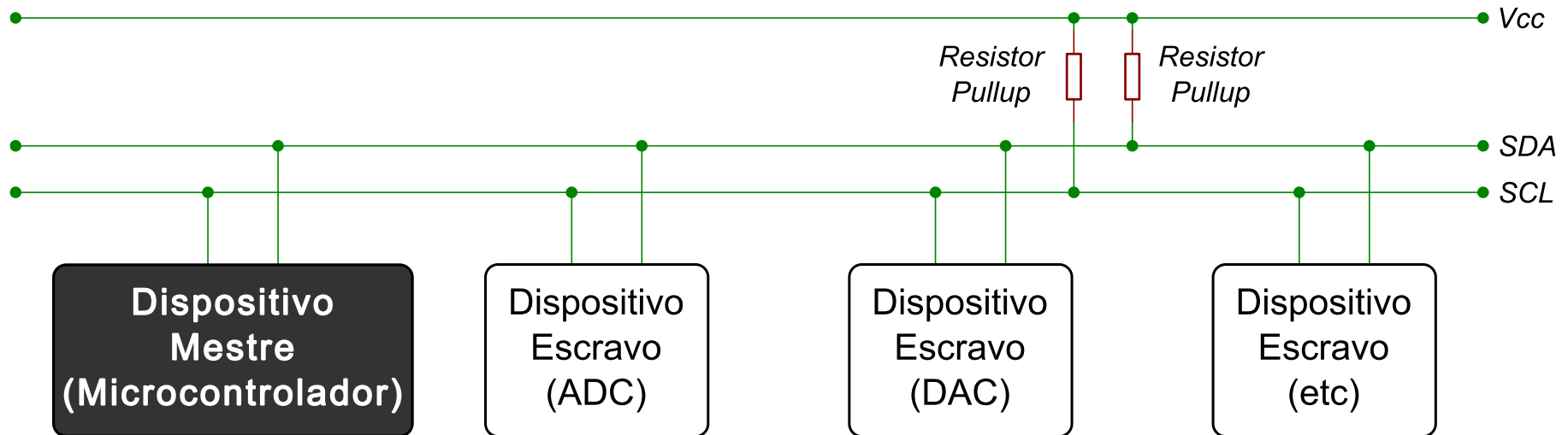
Receptor GPS

# Mensagem do GPRMC

- Formada por 14 campos:
  - 0 - ID da mensagem
  - 1 - Tempo (UTC)
  - 2 - Rastreando (R) ou Aceitável (A)
  - 3 - Latitude
  - 4 - Norte/Sul
  - 5 - Longitude
  - 6 - Leste/Oeste (E/W)
  - 7 - Velocidade (magnitude)
  - 8 - Velocidade (ângulo)
  - 9 - Data (UTC)
  - 10 - Variação Magnética (ângulo em graus)
  - 11 - Variação Magnética (direção, E/W)
  - 12 - Modo de operação (N não válido, A autônomo, D diferencial, E estimado, M manual, S simulação)
  - 13 - Checksum

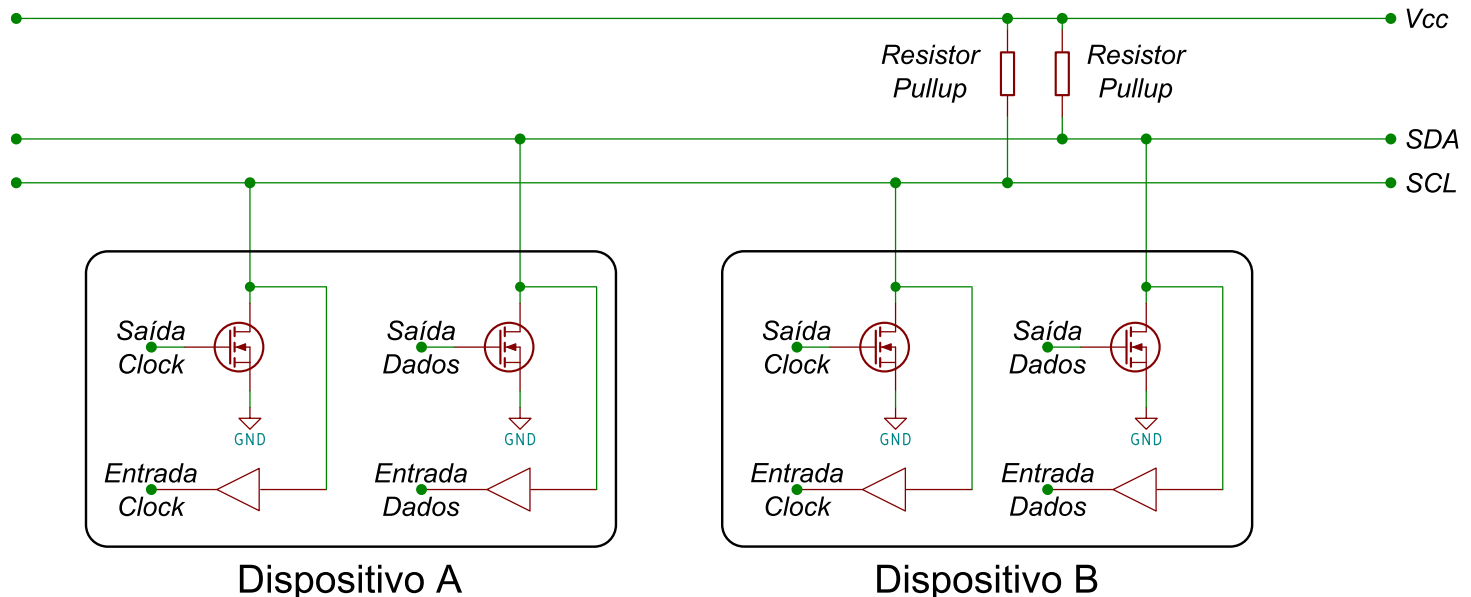
# Comunicação serial I2C

- É um protocolo serial, síncrono e multi-mestre



# Comunicação serial I2C

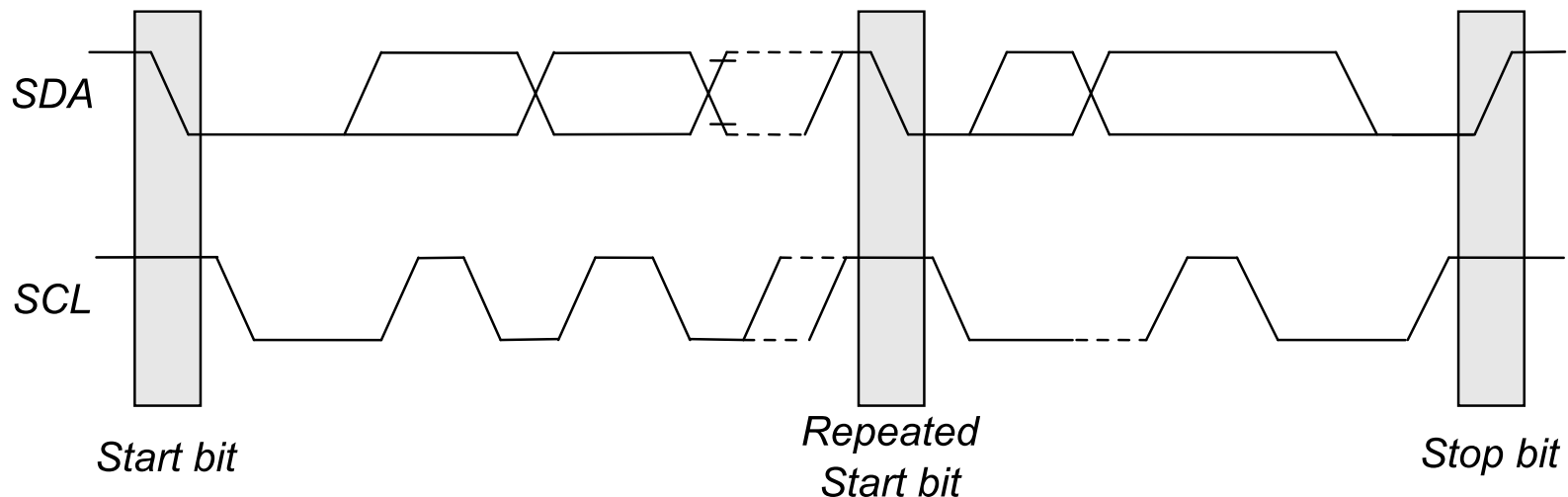
- A comunicação I2C é feita de modo síncrono: os bits são enviados de acordo com um sinal de clock.
- Para isso são necessários pelo menos 2 fios:
  - Transmissão dos dados, SDA ou serial data
  - Transmissão do clock, SCL ou serial clock
- Os terminais são implementados como coletor aberto, exigindo um pull-up para funcionamento.



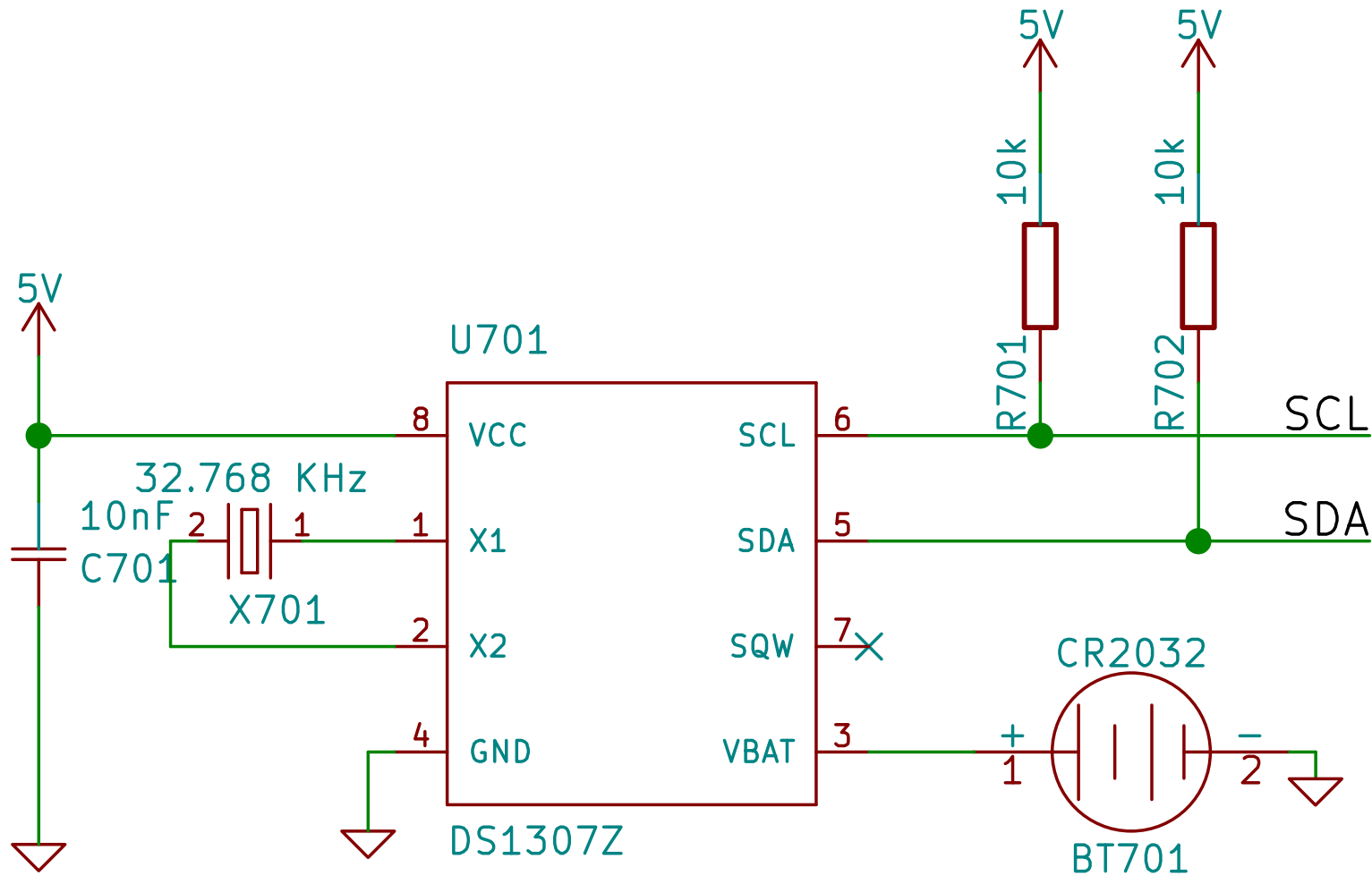


# Comunicação serial I2C

- Existem apenas 2 condições onde o sinal de dados pode mudar seu valor **enquanto** o sinal de clock está em nível alto
  - Start e stop bits
  - O bit de repeated start é uma condição especial do start bit

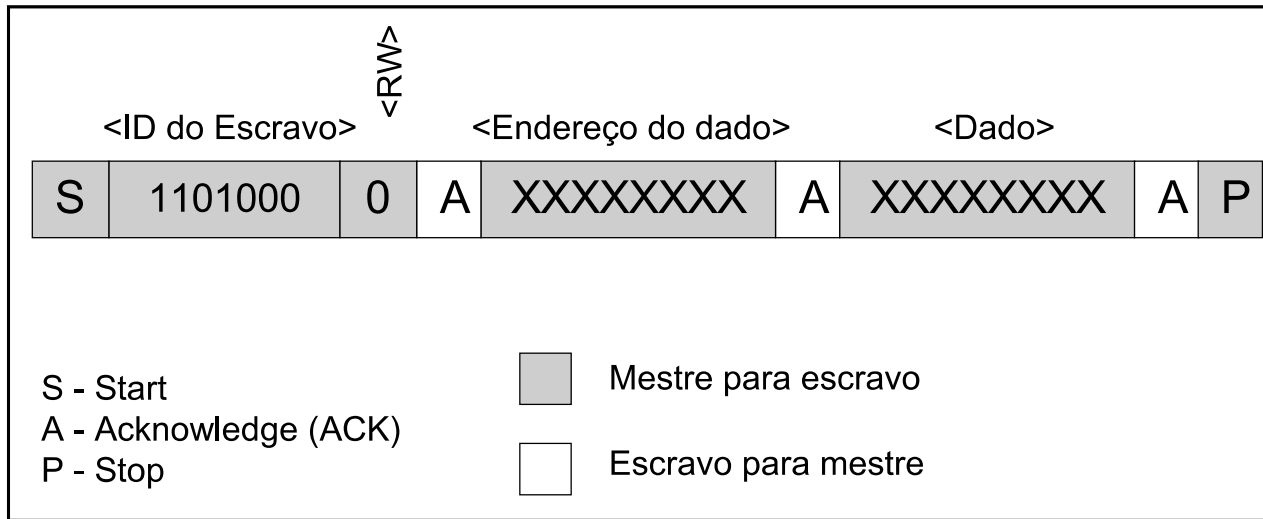


# Interface com dispositivo via I2C

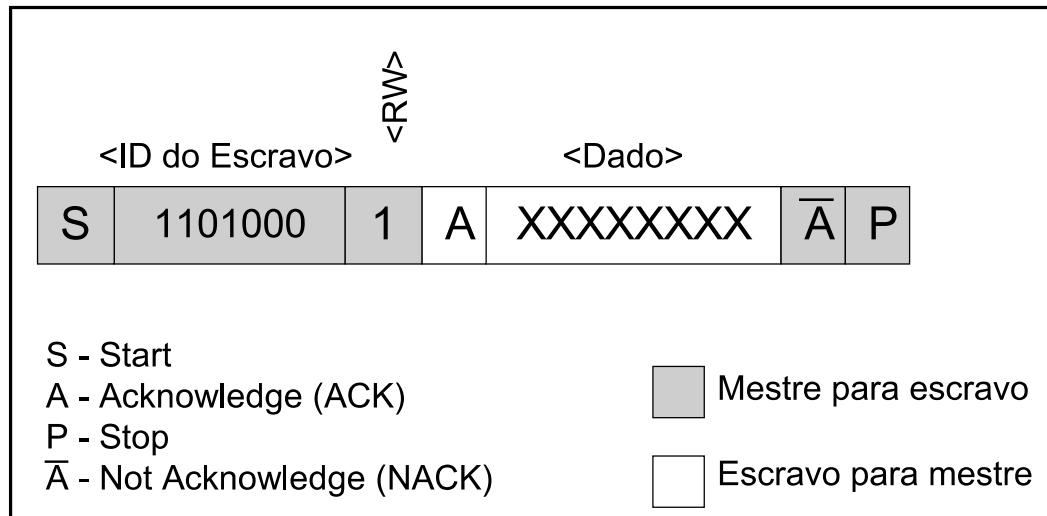


# Comunicando através do I2C

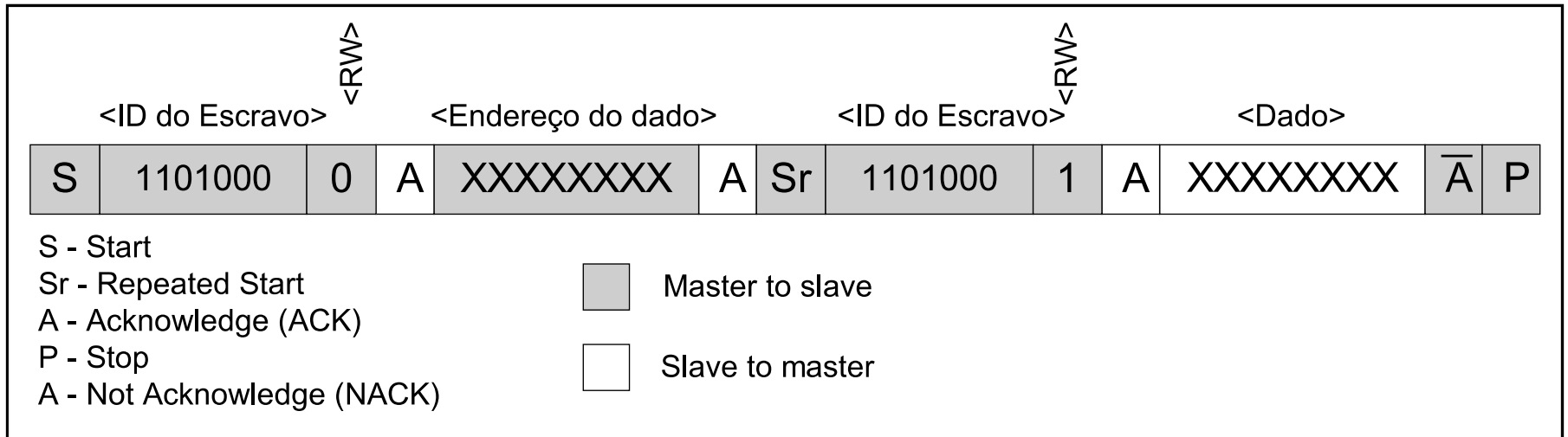
## *Escrita de dados*



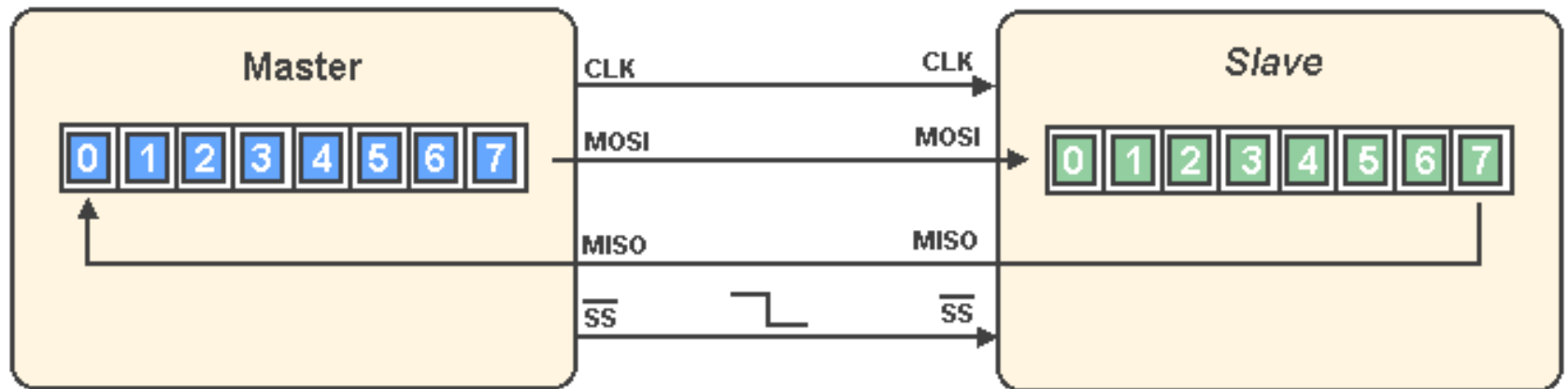
## *Leitura de dados*



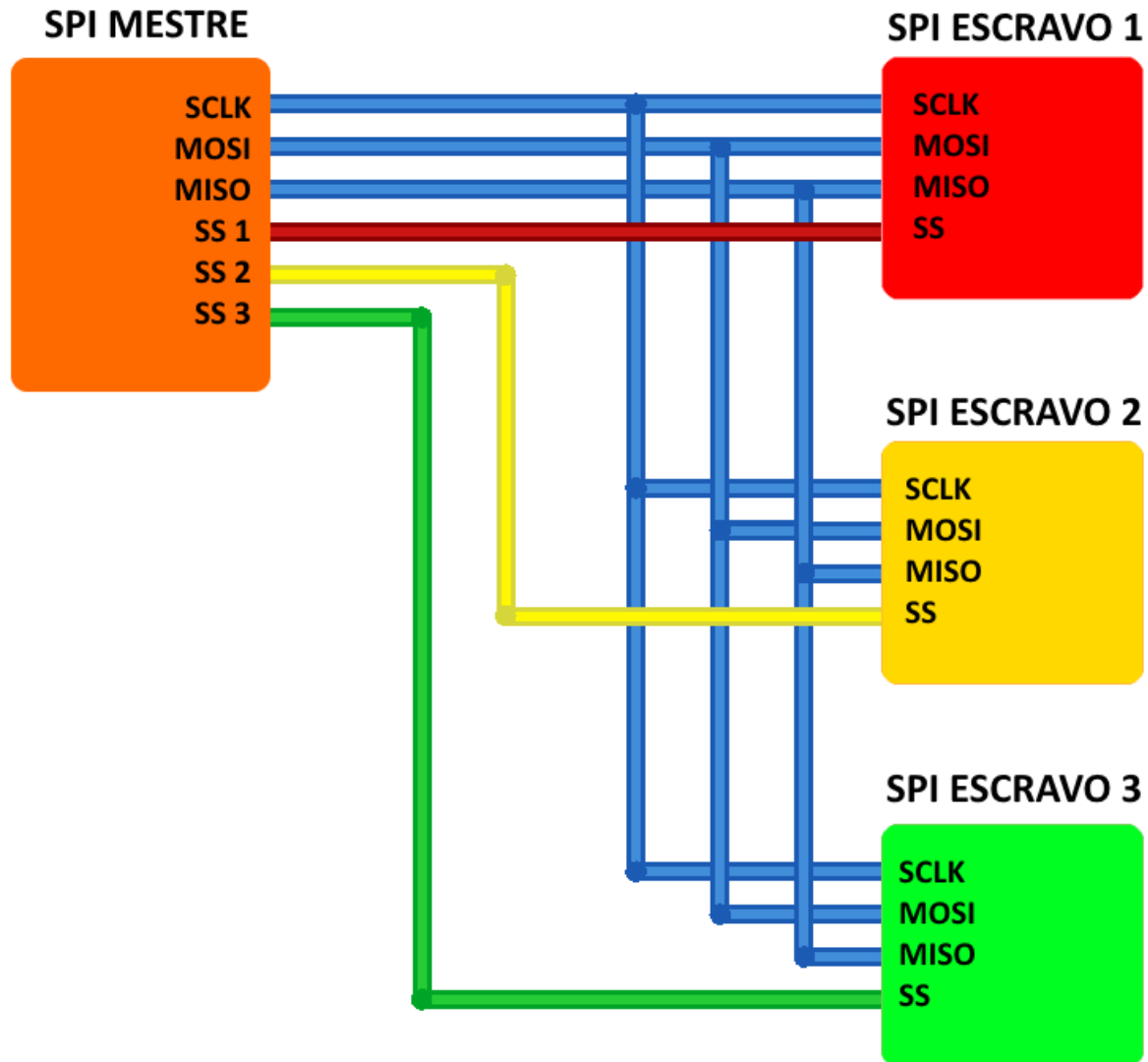
## *Leitura de um endereço específico*



# Interface SPI

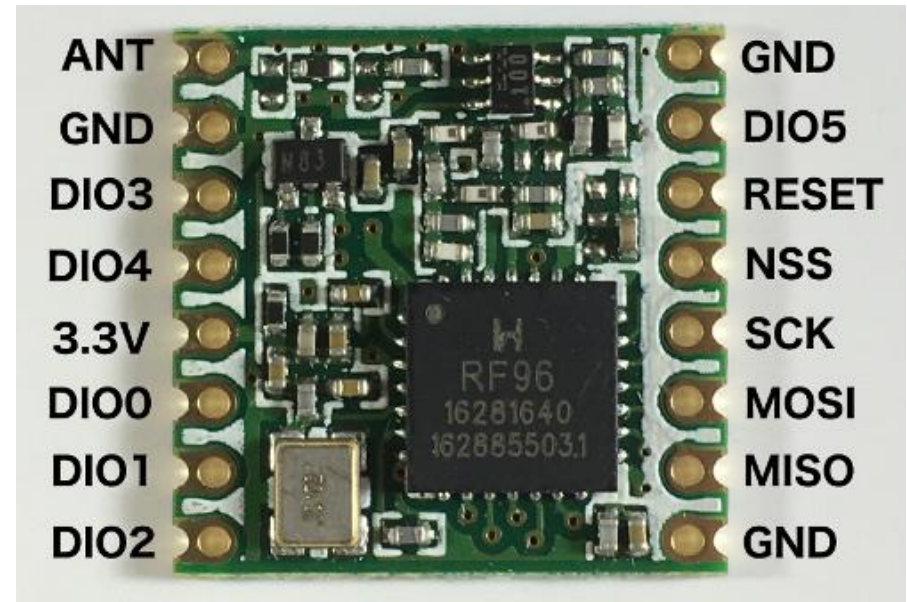
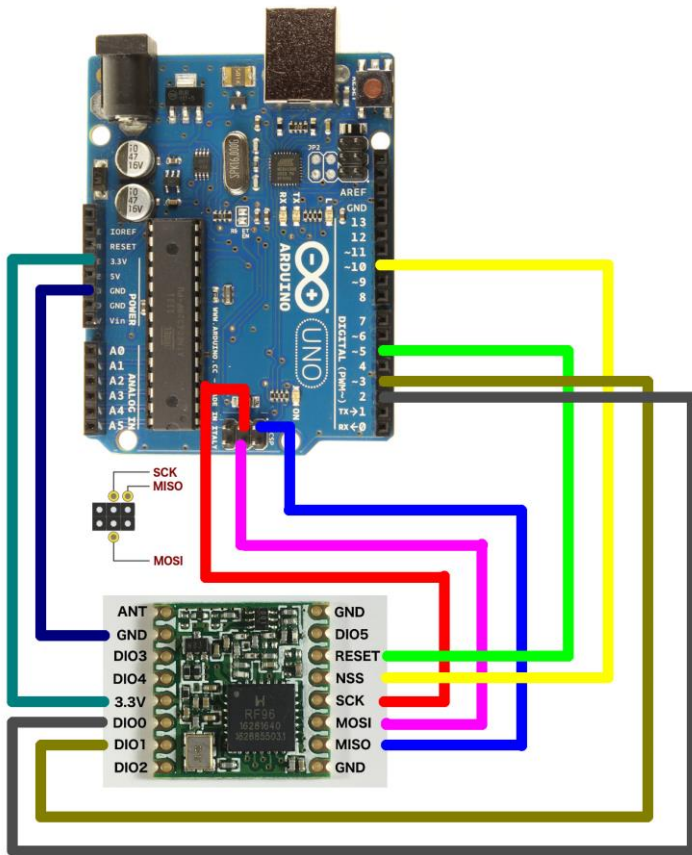


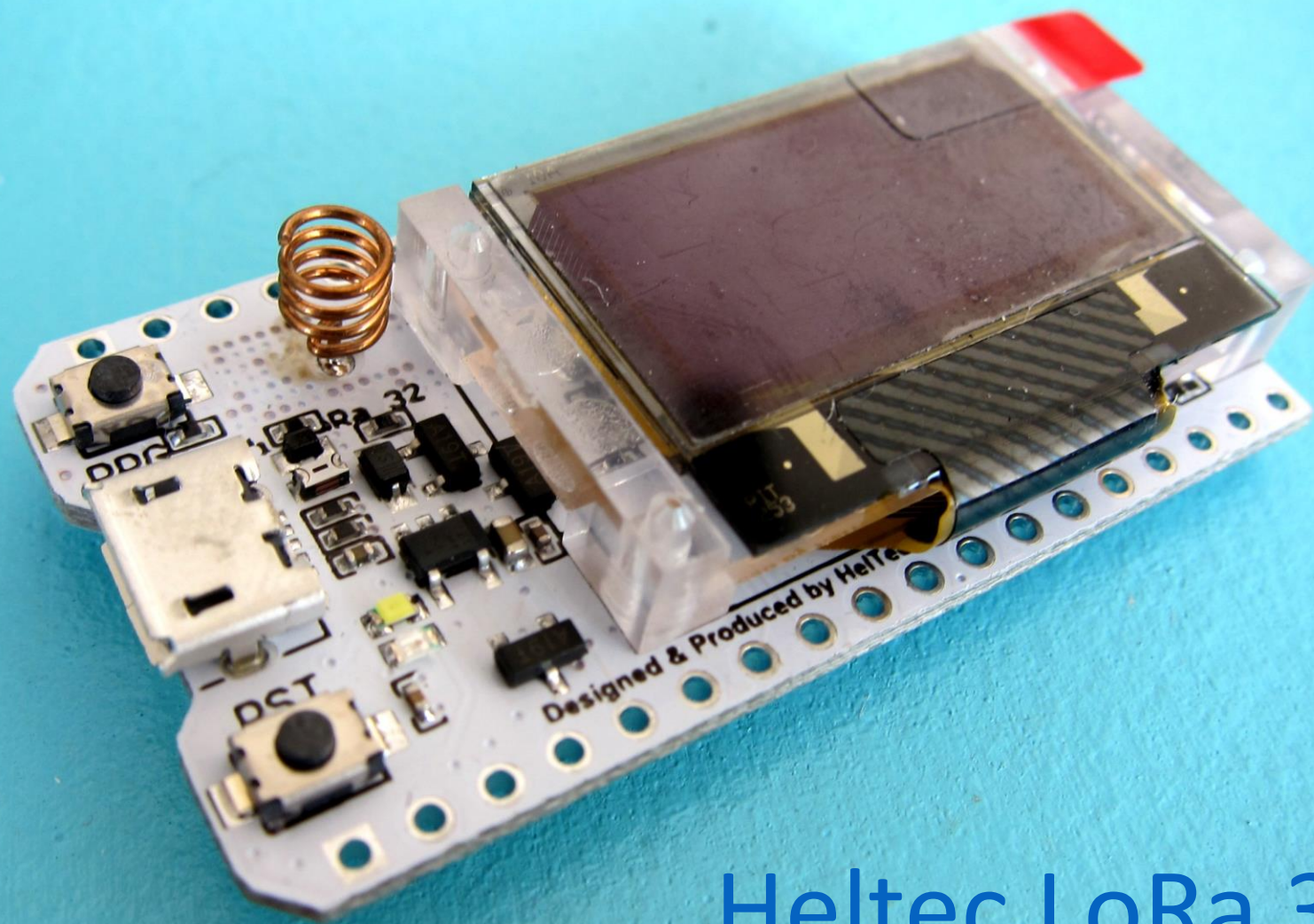
# Interface SPI



# Interface SPI

## Transceiver RFM95 (LORA)





Heltec LoRa 32





ESP32 CAM