

# Comunicação e Sensores com Arduino

Conhecendo o Arduino mais a fundo.



# Luiz Gabriel

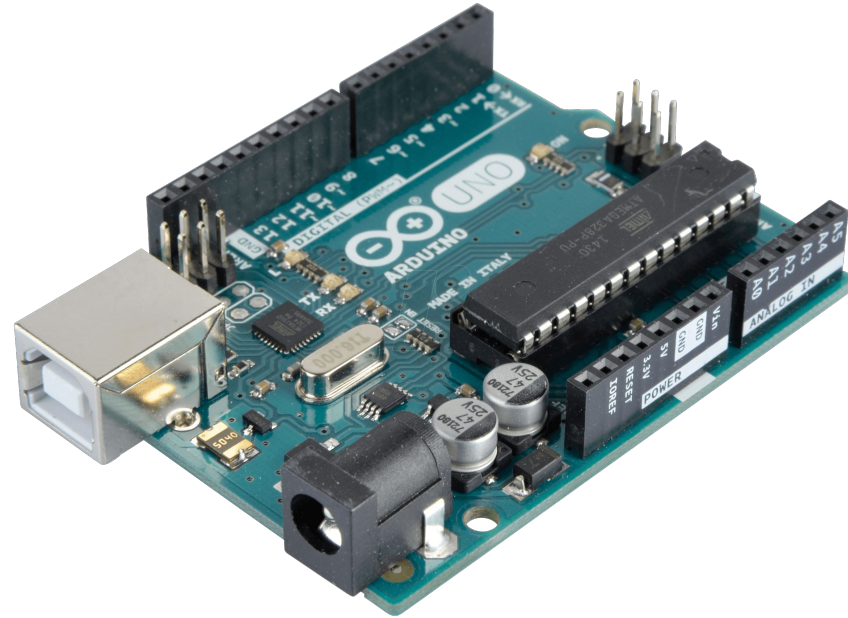
- Formado em Técnico em Informática - IFCE
- Cursando Eng. de Computação - IFCE
- Trabalho como Engenheiro de Software
  - Urbbox Desenvolvimento de Softwares
  - Bynd Caronas Corporativas
- Entusiasta na Área de Microcontroladores, Microcomputadores e Sistemas Embarcados





# Arduino

Uma placa composta por um microcontrolador Atmel, circuitos de entrada/saída e que pode ser facilmente conectada à um computador utilizando C/C++, sem a necessidade de equipamentos extras além de um cabo USB.

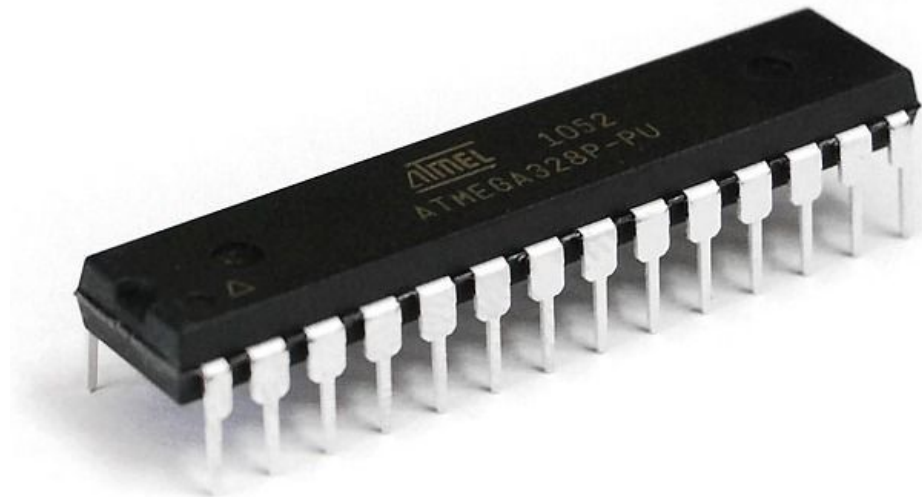




O MICROCONTROLADOR

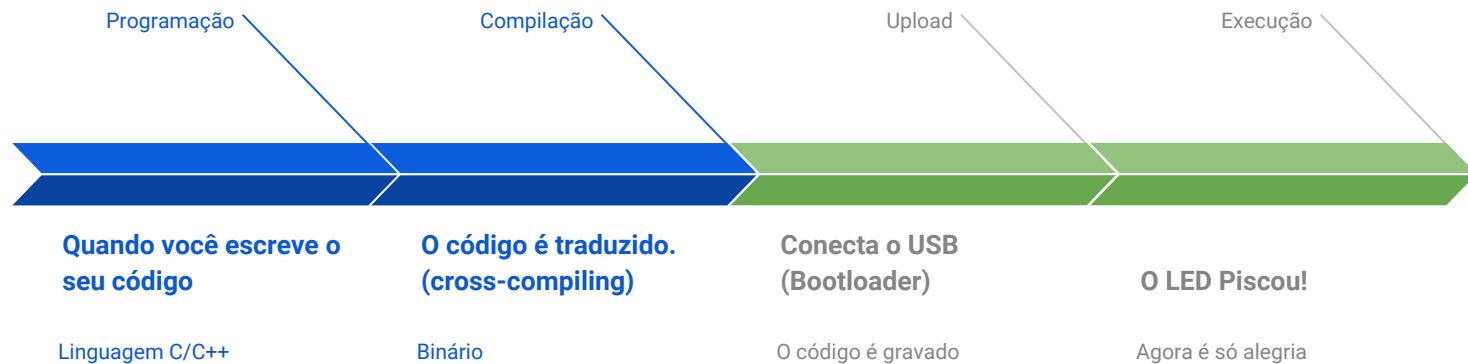
# Atmel AVR

Quando você desenvolve um projeto no Arduino IDE, você está gerando um código que será compilado para o Microcontrolador específico do seu modelo de Arduino.





# O Processo





# PlatformIO is an open source ecosystem for IoT development

Cross-platform IDE and unified debugger. Remote unit testing and firmware updates



28

Platforms



17

Frameworks



552

Boards



173

Examples



5,818

Libraries



Install PlatformIO Now

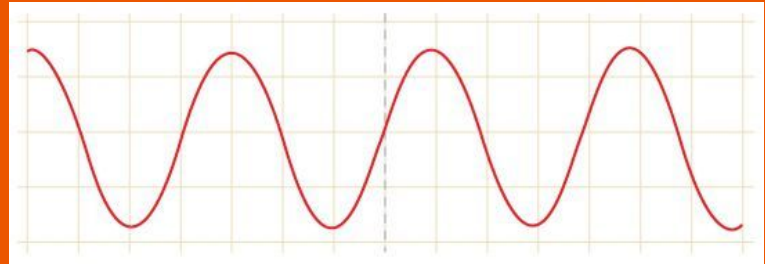
[GitHub](#) · [Bintray](#) · [Twitter](#) · [Facebook](#) · [Hackaday](#) · [Forums](#)

[Release Notes](#) · [Documentation](#) · Sponsored with ❤ by [PIO Plus](#)



# Sinais Analógicos

Como utilizar

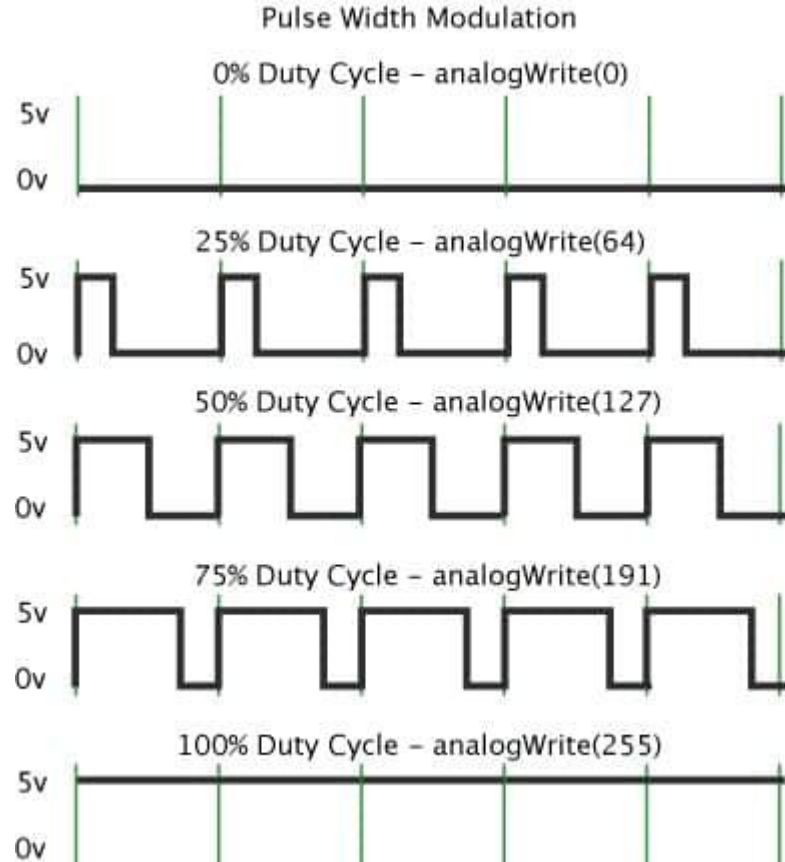


---

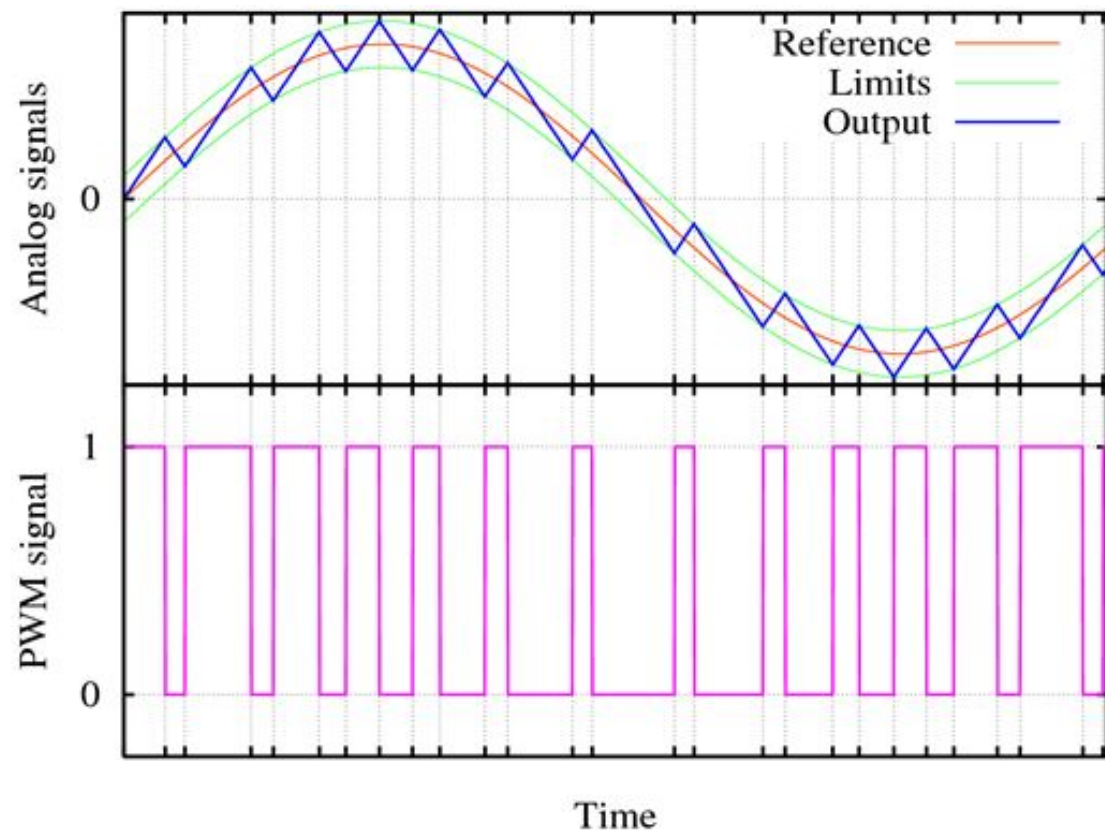


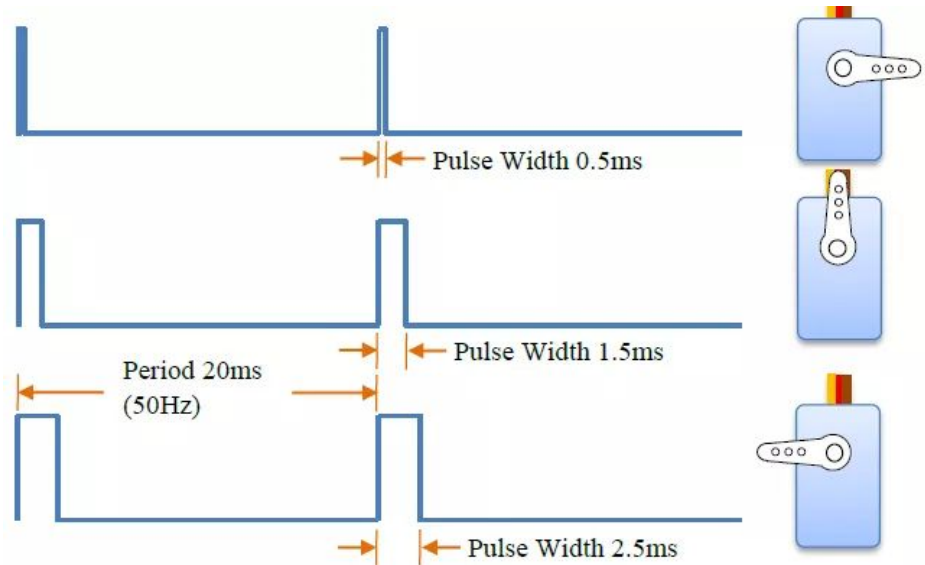
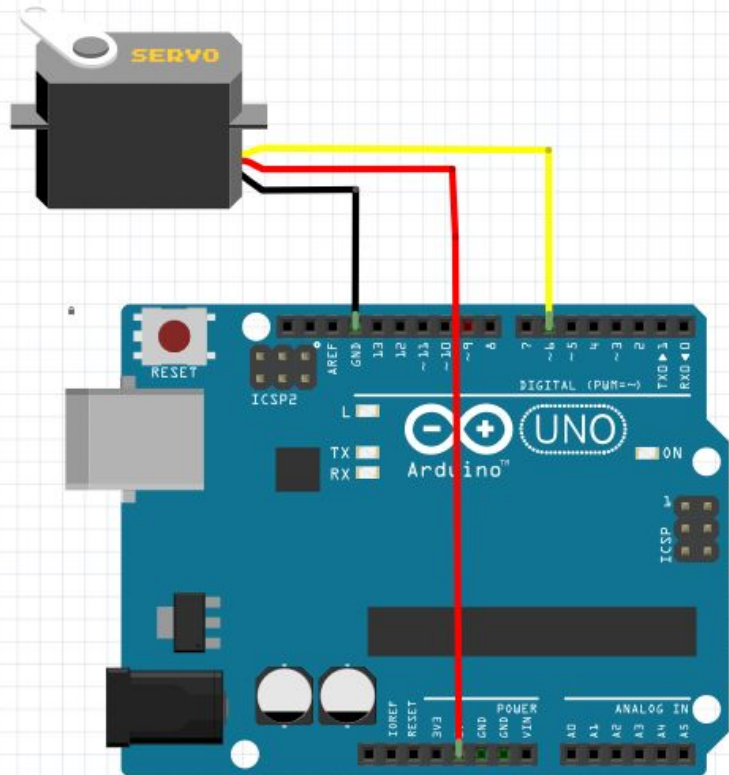
# PWM

Pulse Width Modulation











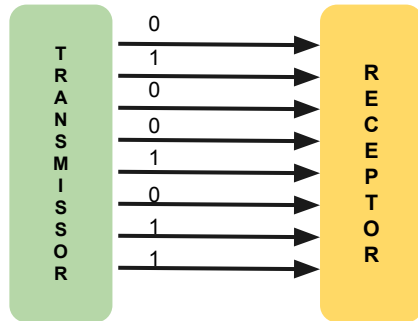
# Comunicação

Teoria e Prática

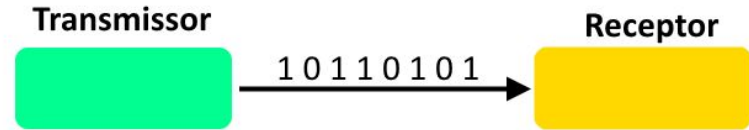




## Transmissão Paralela

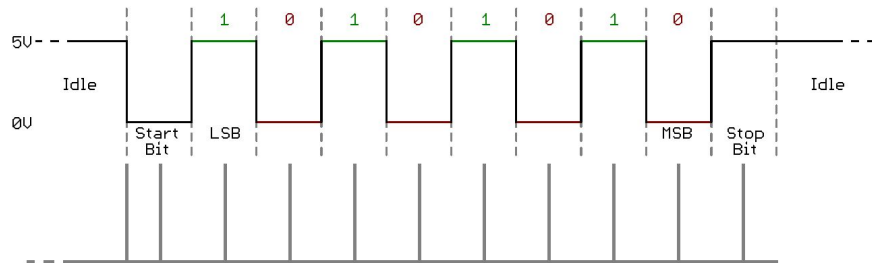
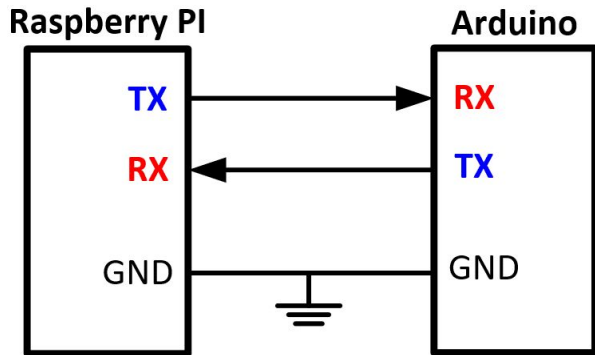


## Transmissão Serial



# USART

Universal synchronous/asynchronous receiver/transmitter

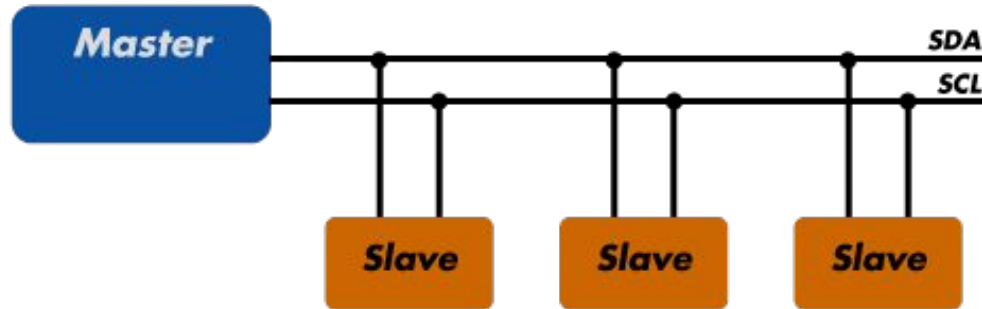


Baud Rate: 1200 / 2400 / 4800 / **9600** / 19200 (bps)  
Taxa de Transmissão (bits/s)

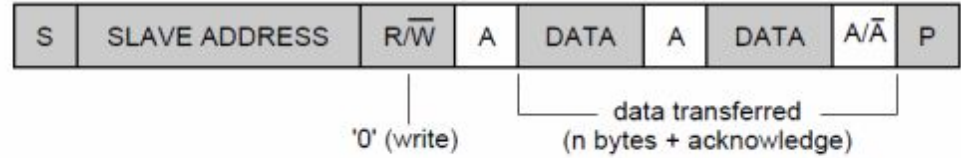


# Inter-Integrated Circuit (I<sup>2</sup>C)

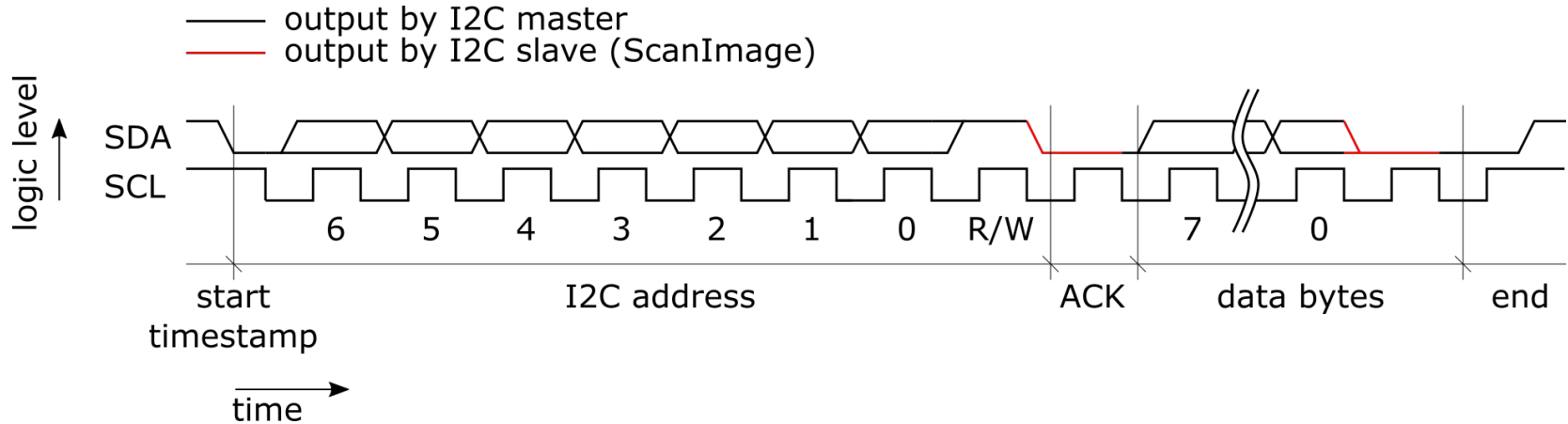
- Desenvolvido pela **Philips** (em 1982)
- Barramento serial Barramento multimestre
- Possível conectar até 127 (usando endereçamento de 8 bits)



# I2C

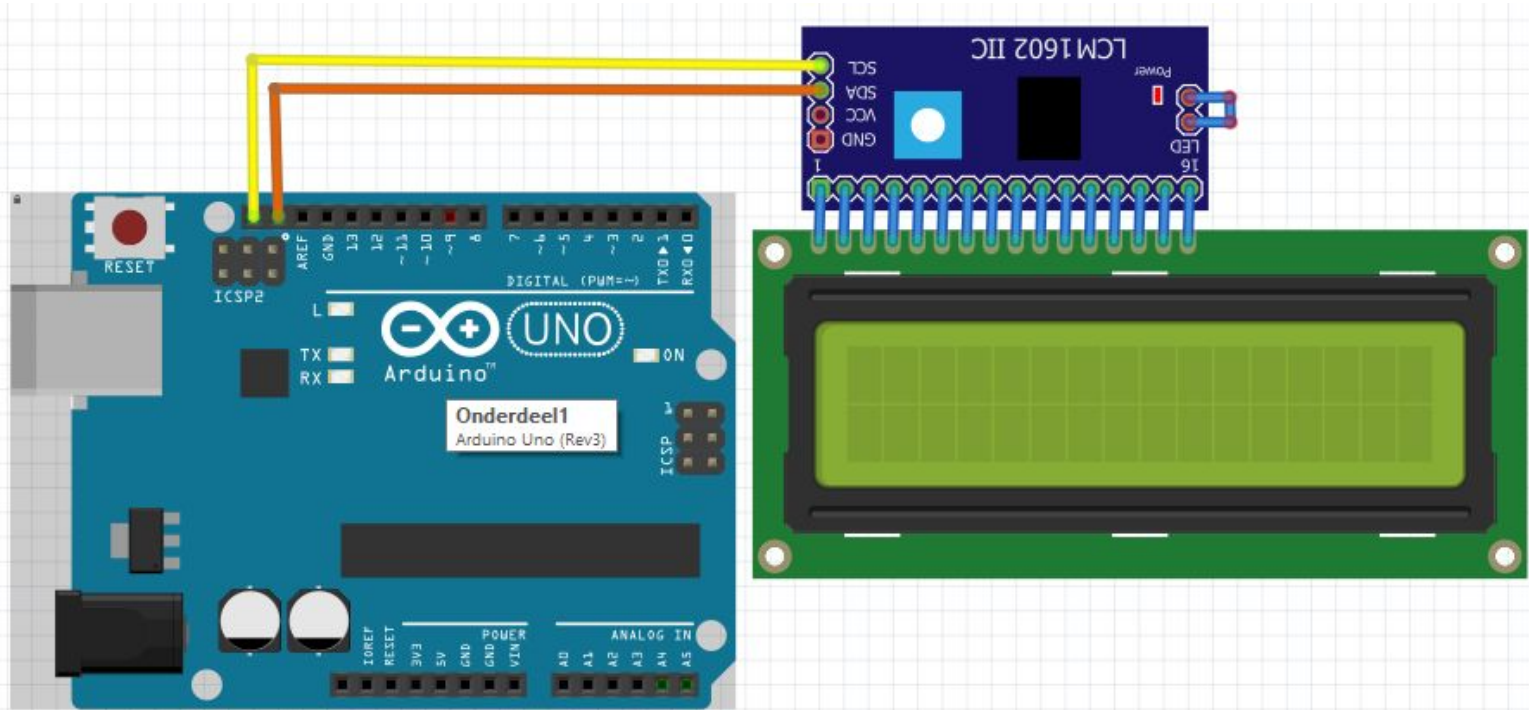


A = acknowledge (SDA LOW)  
 $\bar{A}$  = not acknowledge (SDA HIGH)  
 S = START condition  
 P = STOP condition



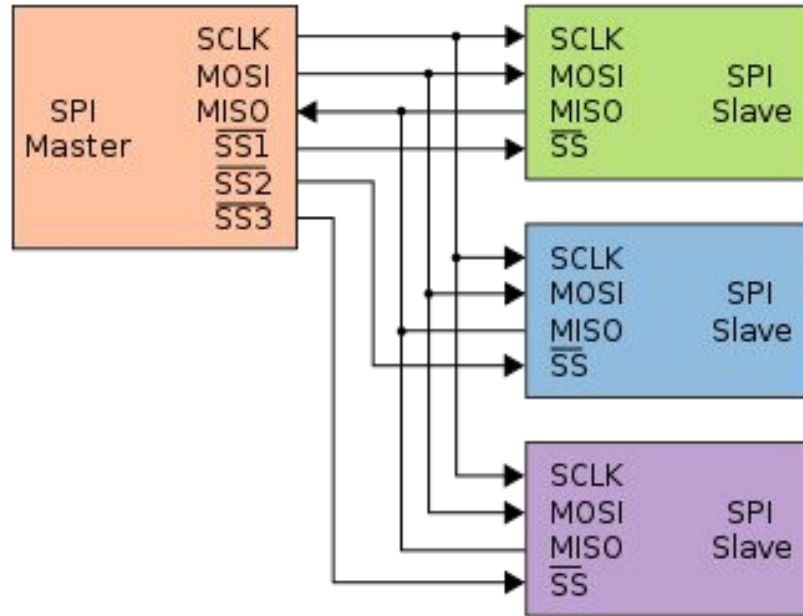


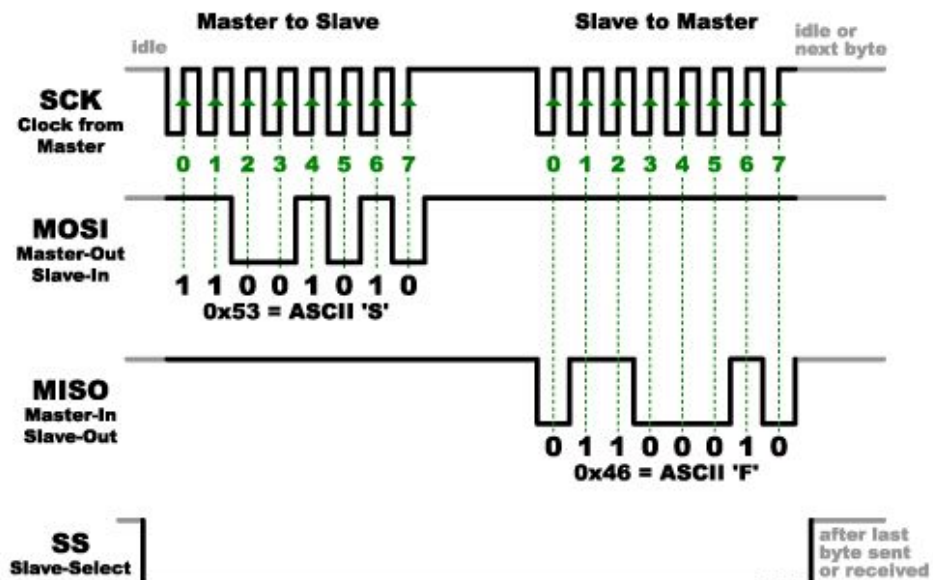
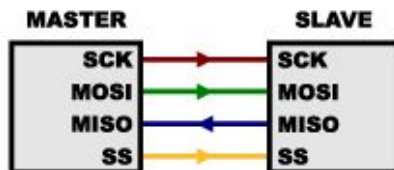
# LCD + I2C



# Serial Peripheral Interface (SPI)

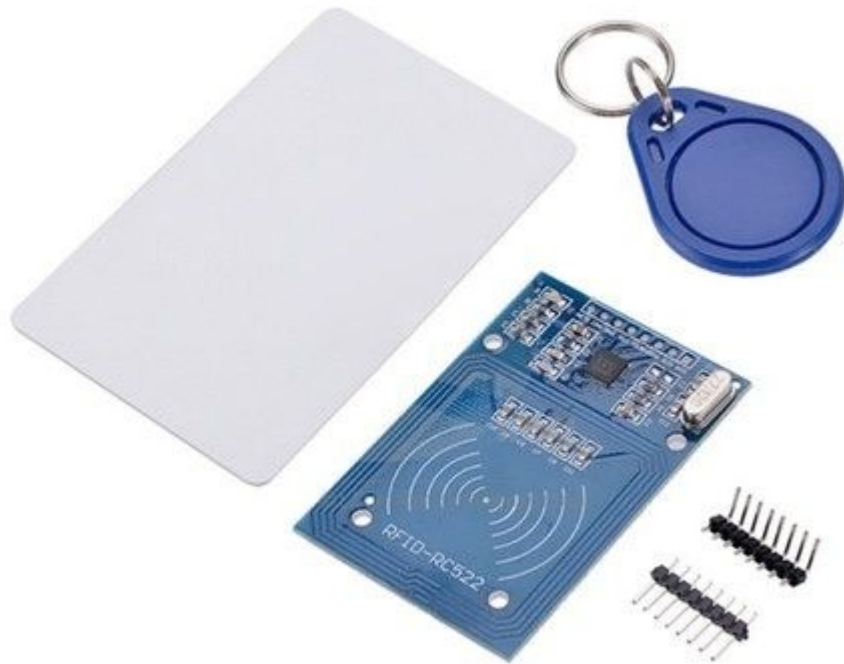
MOSI : Master-Out Slave-In  
MISO : Master-In Slave-Out  
SS : Slave Select

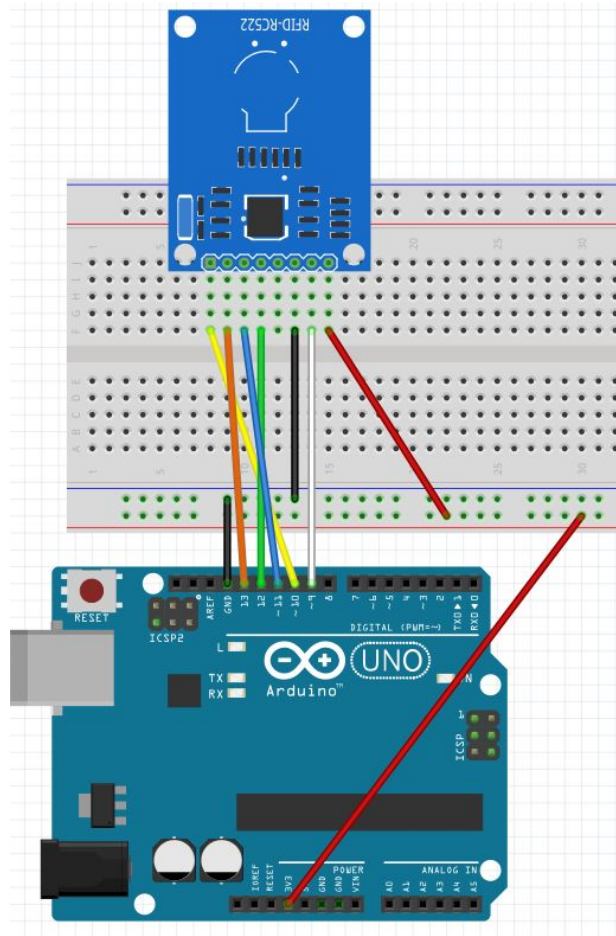




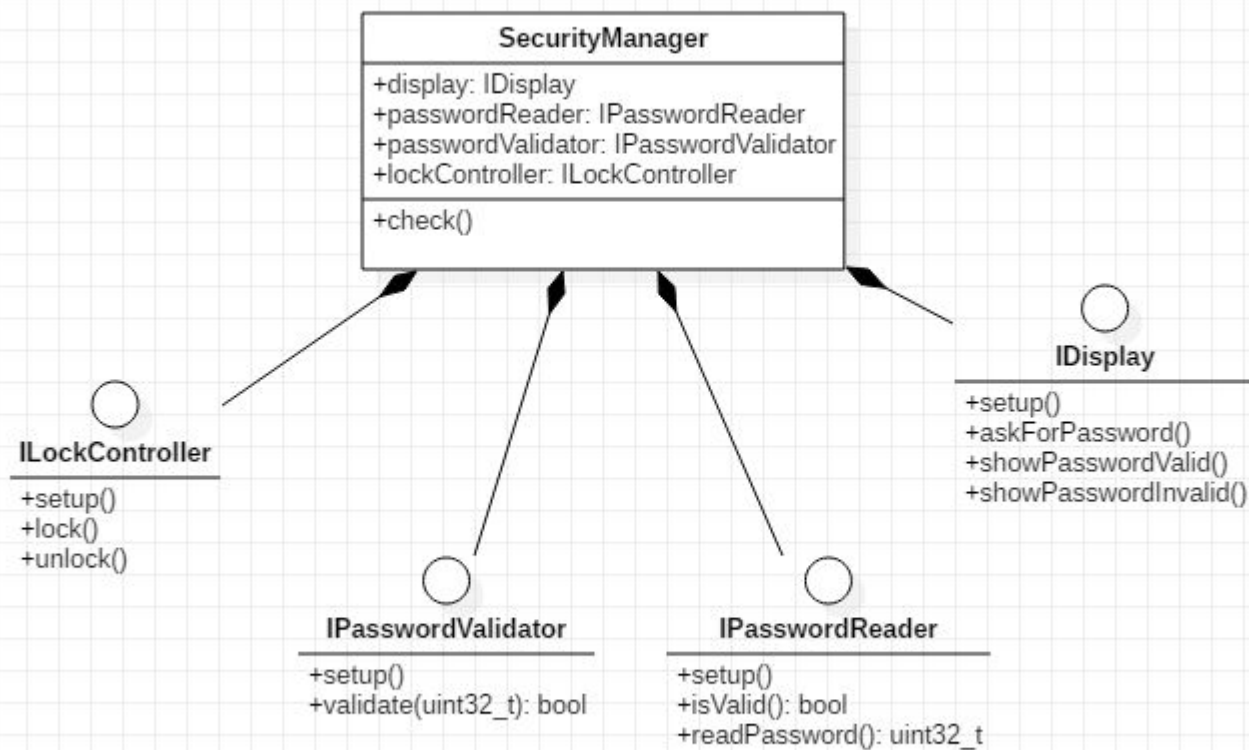
# MFRC522

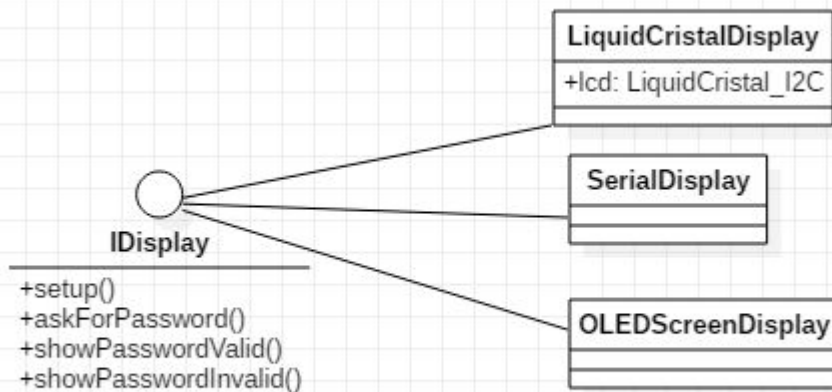
Módulo RFID





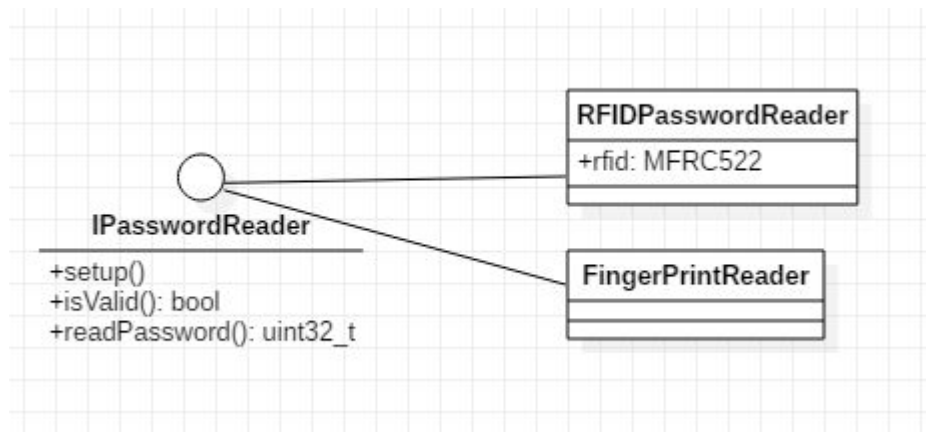
Arduino / Genuino Board	MOSI	MISO	SCK	SS (slave)	SS (master)	Level
Uno or Duemilanove	11 or ICSP- 4	12 or ICSP- 1	13 or ICSP- 3	10	-	5V
Mega1280 or Mega2560	51 or ICSP- 4	50 or ICSP- 1	52 or ICSP- 3	53	-	5V
Leonardo	ICSP- 4	ICSP- 1	ICSP- 3	-	-	5V
Due	ICSP- 4	ICSP- 1	ICSP- 3	-	4, 10, 52	3,3V
Zero	ICSP- 4	ICSP- 1	ICSP- 3	-	-	3,3V
101	11 or ICSP- 4	12 or ICSP- 1	13 or ICSP- 3	10	10	3,3V
MKR1000	8	10	9	-	-	3,3V



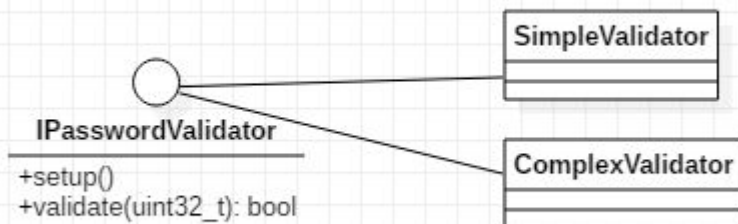


```
class IDisplay
{
public:
    virtual void setup();
    virtual void askForPassword() = 0;
    virtual void showPasswordValid() = 0;
    virtual void showPasswordInvalid() = 0;
};
```

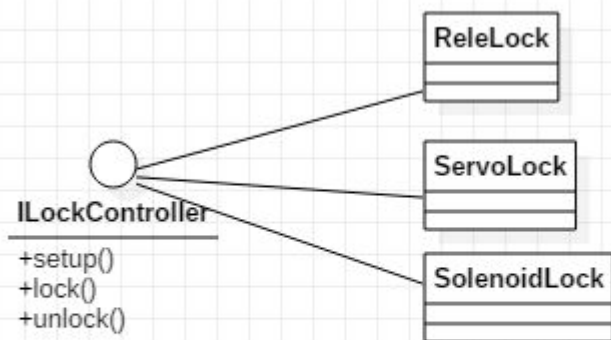




```
class IPasswrodReader
{
public:
    virtual void setup();
    virtual bool isValid() = 0;
    virtual uint32_t readPassword() = 0;
};
```



```
class IPasswordValidator
{
public:
    virtual void setup();
    virtual bool validate(uint32_t password) = 0;
};
```



```
class ILockController
{
    public:
        virtual void setup();
        virtual void lock() = 0;
        virtual void unlock() = 0;
};
```

```
class SecurityManager
{
public:
    SecurityManager();
    void setDisplay(IDisplay *display);
    void setPasswordReader(IPasswordReader *passwordReader);
    void setLockController(ILockController *lock);
    void setPasswordValidator(IPasswordValidator *validator);
    void check();

private:
    IDisplay *m_display;
    IPasswordReader *m_passwordReader;
    ILockController *m_lock;
    IPasswordValidator *m_passwordValidator;
};
```

```
void SecurityManager::check()
{
    if (!this->m_passwordReader->isValid())
        return;

    uint32_t password = this->m_passwordReader->readPassword();

    if (this->m_passwordValidator->validate(password)) {
        this->m_display->showPasswordValid();
        this->m_lock->unlock();
    } else {
        this->m_display->showPasswordInvalid();
        delay(2000);
    }

    this->m_display->askForPassword();
    this->m_lock->lock();
}
```



Link do Projeto no Github:

<https://github.com/luizgabriel/WeekComp.SecuritySystem>



# Obrigado!



Gabriel Lacerda



[github.com/luizgabriel](https://github.com/luizgabriel)



[@luizgabriel1298](https://www.instagram.com/luizgabriel1298)