Tema 4

Módulo 1

**Bibliotecas e Ecossistema do Arduíno:**

Uma das principais características do Arduino é que todas **as suas placas são** **de** **código aberto**, possibilitando, assim, que os usuários as criem de forma independente e, se quiserem, adaptem-nas às suas necessidades particulares. **O software também é de código aberto** e possui uma comunidade de desenvolvedores ativa que disponibiliza projetos e tira dúvidas em fóruns on-line.

\*O fato de ter se originado com fins educacionais tornou o software Arduino bastante atrativo para iniciantes e com muito potencial para usuários avançados. Ele é **multiplataforma**, o que significa que pode operar em Mac, Windows e Linux.

Controlador:

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Vantagens do Arduíno:

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Tela de celular com publicação numa rede social

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Ecossistema arduíno:

**A placa Arduino:**  é nela que o código que implementamos será executado. Semelhante ao que ocorre com um computador, a placa precisa ser conectada à eletricidade. Em seguida, para desenvolver projetos que interajam com o ambiente, é necessário incluir componentes específicos.

**IDE Arduino**: podemos usar para criarmos um programa Arduino e carregá-lo na placa. O fato de estarmos desenvolvendo projetos físicos influencia algumas questões de nomenclatura no Arduino. Uma delas é a forma como um programa Arduino é denominado: esboço.

**Linguagem Arduino**: Quando instalamos a IDE do Arduino, também temos acesso à biblioteca principal do Arduino, que traz um conjunto de funções que nos permite controlar os vários aspectos da funcionalidade da placa.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

Unidades lógicas do Arduíno:

Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Site

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

**Bibliotecas:**

São uma coleção de códigos que facilitam a interação com sensores, display e diversos outros dispositivos, como podemos ver no site oficial do Arduino (Libraries).

As bibliotecas do Arduino são gerenciadas em três locais diferentes:

(1) na opção “Manage Libraries”, (2) em “Add .ZIP Library” e (3) “Arduino Libraries”.

\*\*\*A recomendação oficial (ARDUINO, 2021) é fazer a instalação das bibliotecas na pasta onde o projeto está sendo desenvolvido, para que não sejam excluídas durante o processo de atualização da IDE do Arduino.

**Shields**:

São placas com fins específicos que podem ser conectadas a uma placa Arduino. Essa conexão é feita, literalmente, empilhando as Shields sobre a placa Arduino e fazendo as devidas conexões.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Componentes**: São itens eletrônicos que precisamos acessar pontualmente para personalizar o modo como ele se integra ao nosso projeto. Exemplos de componentes são: teclado, uma tela LCD de caracteres, sensores de diversos tipos (movimento, luz, temperatura, entre outros), circuitos integrados de lógica binária, circuitos amplificadores, scanners de impressão digital, entre tantos outros componentes. Além disso, o Arduino ainda tem os chamados componentes discretos, tais como resistores, diodos, transistores, led, relays, capacitores e indutores.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Bibliotecas

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

Pyserial:

Utilizado para quando se quer usar Python no lugar de C/C++ para utilizar os comandos no Arduíno.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Módulo 2

**NODEMCU**

Ele é baseado na família do microcontrolador ESP8266 que possui recursos que facilitam trabalhar com dispositivos conectados à Internet para monitoramento e controle.

* Tensão operacional: 3,3V.
* Consumo de corrente: 10uA-170mA.
* Memória flash acoplável máx. 16MB (512 K normal).
* Processador Tensilica: L106 de 32 bits.
* Velocidade do processador: 80-160MHz.
* RAM: 32K + 80K.
* GPIOs: 17 (multiplexado com outras funções).
* Uma entrada analógica para digital com 1024 níveis (10 bits) de resolução.
* Wi-Fi: suporte a 802.11 b/g/n.
* Máximo de 5 conexões TCP simultâneas.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**CONFIGURAÇÃO DA PLACA NODEMCU**

A primeira e mais importante questão a qual precisamos ficar atentos é sobre a comunicação serial que é necessária para a maioria dos dispositivos e para testar e corrigir os programas.

\*\*A comunicação serial é usada para habilitar que o microcontrolador ESP8266 se comunique com o computador e com dispositivos de comunicação serial. Todas as linhas de entrada e saída (I/O) de tensão de alimentação para o microcontrolador ESP8266 são de 3,3V (ESP8266, 2021). Por isso, não devemos conectar nenhuma linha I/O com lógica de 5V.

Tabela

O conteúdo gerado por IA pode estar incorreto.

* **Pinos de alimentação**: existem quatro pinos de alimentação: um pino VIN que pode ser usado para alimentar diretamente o ESP8266 e seus periféricos, caso usemos uma fonte de tensão regulada de 5V e três pinos de 3,3V que são a saída de um regulador de voltagem. Esses pinos podem ser usados para fornecer energia a componentes externos.
* **GND**: é um pino terra da placa do ESP8266 NodeMCU.
* **Os pinos I2C**: são usados para conectar todos os tipos de sensores e periféricos I2C nos projetos que viermos a desenvolver. Suporta tanto I2C Master, como I2C Slave.
* **Pinos GPIO**: são os pinos de entrada e saída de propósito geral. O ESP8266 NodeMCU tem 17 pinos GPIO que podem ser atribuídos a várias funções, de forma programática, como I2C, I2S, UART, PWM, controle remoto infravermelho, luz de LED e botão.
* **Canal ADC**: é o conversor analógico-digital (ADC) do NodeMCU.
* **Pinos UART (universal asynchronous receiver transmitter)**: ESP8266 NodeMCU tem 2 interfaces UART, ou seja, UART0 e UART1, que fornecem comunicação assíncrona (RS232 e RS485), e pode se comunicar em até 4,5Mbps. UART0 (pinos TXD0, RXD0, RST0 e CTS0) pode ser usado para comunicação.
* **Pinos SPI**: O ESP8266 possui dois pinos SPIs (SPI e HSPI) nos modos escravo e mestre. Essas SPIs também oferecem suporte aos seguintes recursos de SPI de uso geral:

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

* **Pinos SDIO**: O ESP8266 tem a Interface de entrada e saída digital segura (SDIO) que é usada para fazer interface direta com cartões SD.
* **Pinos PWM**: a placa possui 4 canais de modulação por largura de pulso (PWM) que pode ser programada para acionar motores digitais e LEDs.
* **Pinos de controle**: são usados para controlar o ESP8266. Esses pinos são:

Texto

O conteúdo gerado por IA pode estar incorreto.

**PROGRAMAÇÃO DA PLACA NODEMCU**

O firmware do NodeMCU é baseado na linguagem de programação LUA. Ele é um código aberto para o ESP8266 WiFi SOC da Espressif e usa um sistema de arquivos SPIFFS que está na memória flash (ver nodemcu/nodemcu-firmware, 2021).

É interessante ver que podemos programar no NodeMCU usando as seguintes linguagens: C, Python, Lua, Basic e Js.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Módulo 3

**RASPBERRY PI**

O Raspberry Pi é um computador de baixo custo do tamanho de um cartão de crédito que se conecta a um monitor de computador ou TV e usa um teclado e mouse padrão (ESP8266, 2021). Nele, podemos programar em linguagens como Scratch e Python e fazer quase as mesmas atividades que um computador desktop comum, como navegar na Internet, assistir a vídeos, criar planilhas, editar texto e jogar.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

**CONFIGURAÇÃO DA RASPBERRY PI**

* **raspi-config**: é utilizado para habilitar recursos como a câmera e alterar configurações como o layout de teclado, por exemplo.
* **config.txt**: é o arquivo de configuração do Raspberry Pi.
* **Rede TCP/IP**: é utilizado para configurar a pilha de rede TCP/IP no Raspberry Pi.
* **Conexão a uma rede sem fio e Ponto de acesso sem fio**: é utilizada para configurar o Raspberry Pi para se conectar a uma rede sem fio por meio da conectividade sem fio embutida nas versões do Raspberry Pi 4 ou Raspberry Pi 3.
* **proxy:** usado para configurar o Raspberry Pi para acessar a internet por meio de um servidor proxy.
* **HDMI Config**: faz a configuração do dispositivo HDMI.
* **Editor de configuração de tela**: configura a resolução, frequência e orientação do dispositivo de exibição do editor gráfico.
* **Configuração de áudio**: alterna a saída de áudio entre HDMI e o conector de 3,5mm. O Raspberry Pi tem até três modos de saída de áudio: HDMI 1 e 2, e um conector de fone de ouvido. Podemos alternar entre esses modos a qualquer momento.
* **Configuração da câmera**: é usado para instalar e configurar a placa da câmera Raspberry Pi.
* **Configuração de armazenamento externo**: faz a montagem e configuração de armazenamento externo em um Raspberry Pi.
* **Localização**: faz a configuração do Raspberry Pi para funcionar no idioma e fuso horário local.
* **Configuração de pinos padrão**: é usado para alterar os estados dos pinos padrão. Segundo a própria fonte oficial (RASPBERRYPI, 2021), deve ser usada apenas por usuários experientes.
* **Configuração de árvores de dispositivos**: o kernel e o firmware do Raspberry Pi usam uma árvore de dispositivos para endereçar o hardware conectado à placa. Essas Árvores de Dispositivos podem incluir parâmetros que fornecem um grau de controle sobre alguns recursos integrados.
* **Linha de comando do kernel**: é definida em um arquivo na partição de inicialização, chamado cmdline.txt. Este é um arquivo de texto que pode ser editado em qualquer editor de texto.
* **Configuração UART (*Universal Asynchronous Receiver Transmitter*)**: existem dois tipos de UART disponíveis no Raspberry Pi: PL011 e mini UART. Ambos são de 3,3V.
* **Ícones de aviso de firmware**: descrição dos ícones de aviso exibidos se o firmware detectar problemas.
* **Códigos de flash de aviso de LED**: são mostrados se houver falhas ao inicializar ou desligar o Raspberry PI. Sob certas circunstâncias, o firmware do Raspberry Pi exibirá um ícone de aviso no visor para indicar um problema. Existem três ícones que podem ser exibidos:
  + **Aviso de subtensão**: se a fonte de alimentação do Raspberry Pi cair abaixo de 4,63V.
  + **Aviso de temperatura excessiva**: se a temperatura estiver entre 80°C e 85°C.
  + **Temperatura superexcessiva**: aviso de temperatura acima de 85°C.
* **Protegendo seu Raspberry Pi**: trata de algumas recomendações para tornar o Raspberry Pi mais seguro, como alteração da senha padrão, alteração do nome de usuário, configurar o comando “sudo” para exigir uma senha, certificar-se de ter as correções de segurança atualizadas, aperfeiçoar a segurança SSH, usar a autenticação baseada em chave e instalar um firewall.
* **Protetor de tela**: quando o Raspberry Pi OS é executado em uma área de trabalho gráfica, ele vai deixar a tela em branco após 10 minutos sem intervenção do usuário, por exemplo, sem movimentos do mouse, ou pressionamento de alguma tecla.
* **A pasta de inicialização**: na instalação básica do Raspberry Pi OS, os arquivos de inicialização são armazenados na primeira partição do cartão SD, que é formatado com o sistema de arquivos FAT. Isso significa que ele pode ser lido em dispositivos Windows, macOS e Linux. Quando o Raspberry Pi é inicializado, ele carrega vários arquivos da pasta de inicialização para iniciar o processador e, em seguida, o kernel do Linux.
* **Sistema de arquivos de rede (NFS)**: permite que possamos compartilhar um diretório localizado em um computador em rede com outros computadores, ou dispositivos na mesma rede. O computador onde o diretório está localizado é chamado de servidor e os computadores ou dispositivos que se conectam a esse servidor são chamados de clientes.

**PROGRAMAÇÃO DO RASPBERRY PI**

Um dos pontos mais fortes do Raspberry Pi, além do baixo custo e dos recursos que oferece para interagir com outros componentes externos, é oferecer a possibilidade de desenvolver aplicações com a linguagem Python por meio do MicroPython.