

Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia Eletrônica

Kit de Desenvolvimento Modular para o Ensino de Sistemas Embarcados

Autor: Erick Antonio Correa dos Reis
Orientador: Prof. Dr. Guillermo Alvarez Bestard

Brasília, DF
2019



Erick Antonio Correa dos Reis

Kit de Desenvolvimento Modular para o Ensino de Sistemas Embarcados

Monografia submetida ao curso de graduação
em Engenharia Eletrônica da Universidade
de Brasília, como requisito parcial para ob-
tenção do Título de Bacharel em Engenharia
Eletrônica.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Guillermo Alvarez Bestard

Brasília, DF

2019

Erick Antonio Correa dos Reis

Kit de Desenvolvimento Modular para o Ensino de Sistemas Embarcados

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 06 de dezembro de 2019.

Prof. Dr. Guillermo Alvarez Bestard
Orientador

Prof. Dr. Leonardo Aguayo
Convidado 1

MSc. Renato Faria Iida
Convidado 2

Brasília, DF
2019

Resumo

O ensino de microcontroladores é uma etapa extremamente importante no desenvolvimento de algumas engenharias, mas essencial para engenharia eletrônica. Facilitar esse processo, utilizando placas conhecidas com shields, é um dos principais objetivos deste trabalho. Apresentar uma visão geral do funcionamento dos microcontroladores e mostrar opções de módulos que estendem suas funcionalidades é como este trabalho se inicia. Além disso, também é feita uma pequena análise de opções já existentes no mercado e soluções que embasaram a construção deste projeto, que em grande maioria são inacessíveis à alunos de forma geral. Em seguida são feitas algumas escolhas que vão integrar as shields desenvolvidas neste trabalho. Como resultados tem-se o conjunto de funcionalidades a serem colocadas no projeto, esquemáticos, modelos de organização do layout, bem como as conexões de cada uma das shields. Por fim também é mostrado cronograma de tarefas que deve ser executado na continuação deste trabalho buscando diminuir os riscos.

Palavras-chave: MSP430, microcontroladores, shields, kit de desenvolvimento.

Lista de ilustrações

Figura 1 – Subsistemas básicos de um microcontrolador	11
Figura 2 – Arquitetura de von Neumann	13
Figura 3 – Arquitetura de Harvard	13
Figura 4 – Tipos de entrada	14
Figura 5 – Tipos de saída	15
Figura 6 – Kit e shield comum no mercado brasileiro.	20
Figura 7 – Opções de LEDs	22
Figura 8 – Opções de displays	22
Figura 9 – Display 7 Segmentos	23
Figura 10 – Relé	26
Figura 11 – TB6612FNG	27
Figura 12 – Conversor RS232	28
Figura 13 – LaunchPad Evaluation Kit.	31
Figura 14 – Modelo da shield de interface	34
Figura 15 – Sensores	35
Figura 16 – Modelo da shield de sensores	36
Figura 17 – ESP32 Wroom 32	37

Lista de tabelas

Tabela 1 – Principais placas de desenvolvimento	19
Tabela 2 – Comparação LEDs	21
Tabela 3 – Comparação Displays	21
Tabela 4 – Características do Sensor de Temperatura e Umidade	23
Tabela 5 – Características do Acelerômetro	24
Tabela 6 – Características do Receptor Infravermelho	25
Tabela 7 – Comparação de drivers para motores de passo.	26
Tabela 8 – Especificação L298N	26
Tabela 9 – Especificação TB6612FNG	27
Tabela 10 – Pinos de comunicação serial.	32
Tabela 11 – Conexões shield de interface	33
Tabela 12 – Conexões shield de sensores.	35
Tabela 13 – Conexões shield atuadores.	37
Tabela 14 – Conexões shield de conectividade.	37
Tabela 15 – Cronograma.	38

Sumário

1	INTRODUÇÃO	8
1.1	Justificativa	8
1.2	Importância	8
1.3	Objetivos	9
1.3.1	Objetivos Específicos	9
1.4	Metodologia	9
1.5	Organização	9
2	REFERENCIAL TEÓRICO	11
2.1	Microcontroladores	11
2.1.1	Hardware dos Microcontroladores	11
2.1.1.1	Unidade de processamento central	12
2.1.1.2	Memória	12
2.1.1.3	Entradas e Saídas	14
2.1.1.4	Interrupções	15
2.1.1.5	Sinal de Relógio	15
2.1.1.6	Temporizadores	16
2.1.1.7	Interfaces de comunicação	16
2.1.1.8	Conversores	16
2.1.2	Linguagens de Programação	17
2.1.3	Programando um Microcontrolador	18
2.1.4	Aplicações	18
2.1.5	Uso da Educação	18
2.2	Opções do Mercado	19
2.2.1	Microcontroladores e Placas	19
2.2.2	Shields	20
2.2.2.1	BoosterPack	20
2.2.3	Módulos e componentes	21
2.2.3.1	Interfaces	21
2.2.3.2	Sensores	23
2.2.3.3	Atuadores	25
2.2.3.4	Conectividade	27
3	DESENVOLVIMENTO	29
3.1	Requisitos	29
3.2	Restrições	29

3.3	Ferramentas	30
3.4	Proposta	30
3.4.1	Placa principal	30
3.4.2	MSP430	30
3.4.2.1	LaunchPad MSP430	31
3.4.3	Informações	32
3.4.3.1	Esquemático	33
3.4.3.2	Modelo PCB	34
3.4.4	Sensores	34
3.4.4.1	Esquemático	35
3.4.4.2	Modelo PCB	36
3.4.5	Atuadores	36
3.4.5.1	Esquemático	36
3.4.6	Conectividade	37
3.5	Cronograma	38
4	CONCLUSÃO	39
	REFERÊNCIAS	40
	ANEXOS	44
	ANEXO A – LAUNCHPAD DEVELOPMENT KIT PINOUT	45
	ANEXO B – ESQUEMÁTICOS	46

1 Introdução

O mercado de microcontroladores tem tido recorde de vendas entre os anos 2010 e 2018 ([LINEBACK, 2019](#)), não se tratando apenas de número de vendas é nítido o quão acessível produtos eletrônicos tem se tornado para o dia a dia das pessoas.

Entender os princípios de funcionamento de um microcontrolador, bem como utilizá-lo em algum projeto, é uma algo básico e necessário para que um aluno de graduação saia minimamente preparado para esse mercado que ainda tem perspectivas de crescimento.

Na Universidade de Brasília, Faculdade Gama, o contato efetivo com microcontroladores acontece apenas no 6º semestre, e como esperado, os alunos têm que elaborar um projeto final no qual é necessário utilizar alguns periféricos, e é nesse ponto que as dificuldades começam a aparecer. Elaborar um projeto que atenda as expectativas da disciplina normalmente evolui a união de alguns módulos e circuitos externos que, muitas vezes, apresentam problemas que envolvem duas áreas: dificuldade de entendimento do funcionamento e operação dos módulos e, principalmente, empecilhos relacionados à conexão e mal contatos com jumpers.

Visando reduzir o tempo gasto resolvendo esses problemas esse trabalho propõe a elaboração de um kit de desenvolvimento modular utilizando a família de microcontroladores MSP430, da Texas Instruments.

1.1 Justificativa

A variedade de módulos e tipos de conexão dificultam o uso de funcionalidade básicas dos microcontroladores, além disso outro ponto que atrapalha não só o uso, mas também o desenvolvimento de protótipo é a falha nas conexões, jumpers e fios, que por sua vez são extremamente suscetíveis à mal contato. A falta de módulos integrados, conhecidos como shields ou boosterpacks, para a plataforma MSP430 torna esse desafio ainda maior.

1.2 Importância

O uso de microcontroladores no mercado é muito amplo, e o aprendizado nessa área não é simples e tem uma longa cadeia de disciplinas, entretanto é muito claro que somente trabalhando em projetos é possível entender as particularidades do seu uso. Por isso uma plataforma que facilite o desenvolvimento é extremamente importante, já que com uma estrutura robusta é possível focar apenas no projeto e não em erros provenientes da montagem e/ou conexões.

1.3 Objetivos

Desenvolver um sistema modular de baixo custo para o ensino de eletrônica e sistemas embarcados, que possua diversas funcionalidades de sensoriamento, acionamento e comunicação.

1.3.1 Objetivos Específicos

- Escolher o microcontrolador e o hardware que servirá de base para os módulos a serem elaborados.
- Listar e definir interfaces, sensores e atuadores que farão parte dos módulos.
- Construir os módulos compatíveis com a plataforma e que sejam economicamente acessíveis para alunos.
- Disponibilizar a documentação e o firmware necessário para a comunicação e interação com os módulos.

1.4 Metodologia

A realização desse trabalho se dará primeiramente entendendo as necessidades que o produto em si deve cumprir, ou seja, quais funções devem ser desempenhadas, este ponto será feito através de pesquisas com o objetivo de identificar funcionalidades mais comuns e que explorem uma gama de características de um microcontrolador, além disso professores que ministram disciplinas da área serão consultados.

Posteriormente será feita uma pesquisa de componentes que cumpram as funcionalidades especificadas, visando também encontrar opções disponíveis no mercado brasileiro, em seguida será planejado a organização dos módulos nas shields buscando trazer a melhor usabilidade.

Por fim um conjunto de códigos serão desenvolvidos para servir como exemplo de utilização das shields desenvolvidas, esta etapa é importante para facilitar a utilização do produto. Além disso também será desenvolvido material de apoio explicando cada uma das shields e como deve ser sua utilização.

1.5 Organização

Este documento está organizado da seguinte maneira:

- Referencial teórico com uma explicação do funcionamento dos microcontroladores de forma geral, projetos com escopo semelhante e, além disso, uma pesquisa sobre pro-

duto disponível no mercado brasileiro que podem contribuir no desenvolvimento do trabalho;

- Desenvolvimento do projeto em si, justificando todas as escolhas e detalhando a organização dos shields em termos de esquemático e layout, bem como ferramentas a serem utilizadas e um cronograma geral das próximas etapas.
- Conclusão com observações importantes desse trabalho.

2 Referencial Teórico

Antes de detalhar o projeto a ser desenvolvido será apresentado um resumo breve sobre a arquitetura dos microcontroladores bem como informações sobre o uso deles. Além disso será apresentada uma visão geral do mercado com relação a disponibilidade de microcontroladores, placas de desenvolvimento, módulos e componentes.

2.1 Microcontroladores

Como base para o trabalho é necessário conhecer o funcionamento de um microcontrolador, [Davies \(2008\)](#) consegue detalhar muito bem suas características e as mais relevantes podem ser encontradas a seguir.

Em [Dinis Gaspar, Espírito Santo e Ribeiro \(2010b\)](#), [Dinis Gaspar, Espírito Santo e Ribeiro \(2010a\)](#), [Dinis Gaspar, Espírito Santo e Ribeiro \(2010c\)](#) também é possível encontrar informações muito claras a respeito especificamente da MSP430, família de microcontroladores que será melhor detalhada posteriormente.

2.1.1 Hardware dos Microcontroladores

De maneira prática, um microcontrolador é um pequeno computador capaz de executar instruções pré-programadas, entretanto é construído em apenas um circuito integrado que contém todos os subsistemas necessários para seu funcionamento, dentre eles a unidade de processamento central (CPU, sigla para Central Processing Unit), a memória e os periféricos programáveis de entrada e saída.

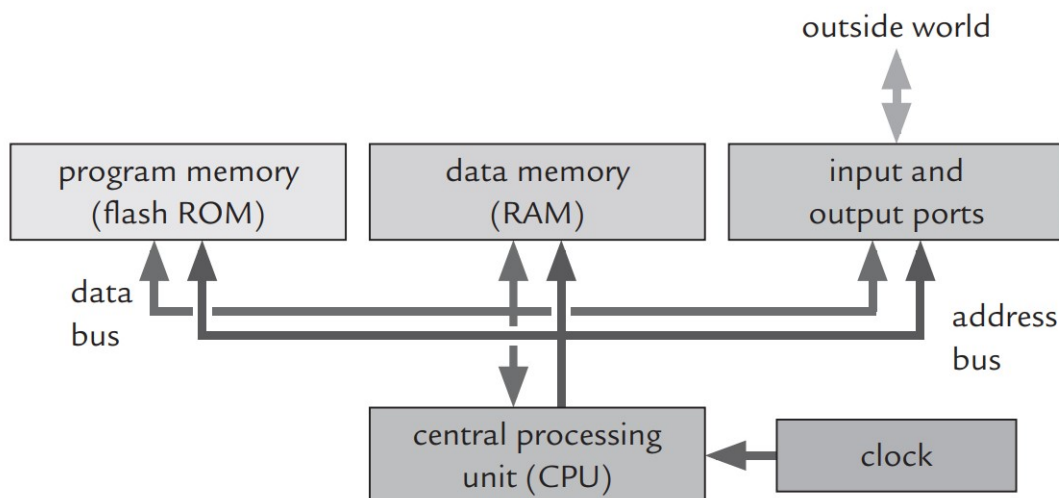


Figura 1 – Subsistemas básicos de um microcontrolador

Fonte: ([DAVIES, 2008](#))

2.1.1.1 Unidade de processamento central

Responsável por executar efetivamente as operações, este subsistema conta com os seguintes componentes:

Unidade lógica aritmética (ULA): Realiza os cálculos necessários nas operações, embora cada microcontrolador possa apresentar diferentes conjuntos de instruções, operações básicas como soma, subtração, AND, OR e deslocamento de bits são alguns dos exemplos de funções que as ALUs podem fazer.

Registadores: Além de fornecerem espaço para as informações básicas de operação da CPU como o program counter (PC), stack pointer (SP) e status register (SR). Além disso, os registradores são responsáveis por armazenar resultados temporários.

Outros: Diversos outros componentes para decodificar instruções, manipular resets e interrupções.

2.1.1.2 Memória

Embora os registradores também sejam responsáveis por armazenar dados, a memória é um componente dedicado exclusivamente para essa função.

RAM

A memória de acesso aleatório (do inglês *Random Access Memory*) normalmente armazenam os dados gerados durante a execução do programa e geralmente são mais rápidas que a ROM, ser do tipo volátil é uma das suas principais características.

ROM

A memória somente de leitura ou ROM (acrônimo em inglês de *read-only memory*), responsável por armazenar o programa gerado. As memórias ROM se encaixam na categoria não-voláteis, pois são capazes de reter as informação mesmo quando a energia é removida.

Para superar a grande limitação de serem apenas para leitura, surgiram diversas opções de memórias programáveis e posteriormente regraváveis como o caso da EEPROM (Electrically-Erasable Programmable Read-Only) e da memória Flash, de longe uma das mais usadas, com funcionamento semelhante às EEPROM, têm uma grande vantagem de velocidade por trabalhar com blocos de memória e não bytes independentes ([HARARI; NORMAN; MEHROTRA, 1994](#)).

Os microcontroladores geralmente utilizam 3 tipos de memória, utilizadas para diferentes funções. Flash contendo o programa em si e EEPROM com dados gerais que

precisam ser salvos por longos períodos e RAM para armazenar dados temporários gerados durante a execução do programa.

Arquiteturas Von Neumann e Harvard

Essas arquiteturas estão relacionadas a organização de memórias do sistema (OLIVEIRA, 2014).

Na arquitetura de von Neumann memórias de dados e de programa compartilham o mesmo barramento, sendo uma arquitetura muito mais simples, porém tem uma desvantagem relacionada a velocidade já que tudo deve passar por um único barramento.

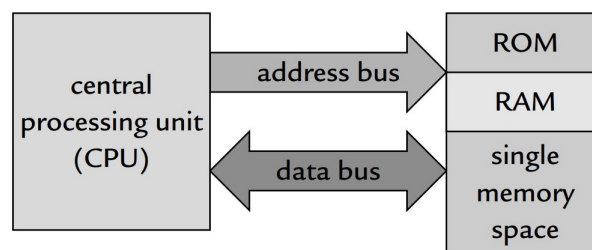


Figura 2 – Arquitetura de von Neumann
Fonte: (DAVIES, 2008)

Já na arquitetura de Harvard essas memórias podem ser acessadas de forma simultânea, pois não compartilham o mesmo barramento, isso torna a comunicação mais rápida, em contrapartida se trata de uma arquitetura mais complexa.

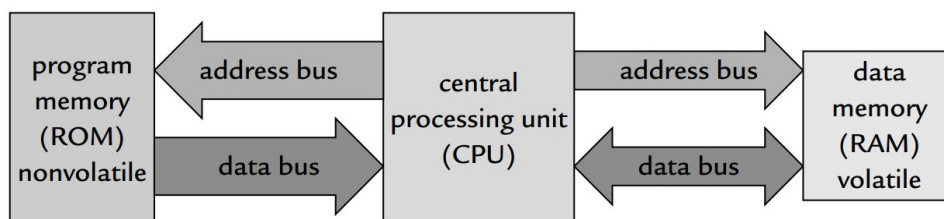


Figura 3 – Arquitetura de Harvard
Fonte: (DAVIES, 2008)

Barramentos de endereço e dados

A comunicação entre todos os componentes acontece a todo momento e a velocidades muito altas, para tornar isso possível é necessário criar essas ligações, que geralmente são separadas entre barramento de endereços, carregando informações a respeito do dispositivo com o qual está se comunicando e barramento de dados, onde efetivamente se trafegam os dados entre dispositivos.

2.1.1.3 Entradas e Saídas

Ponto fundamental dos microcontroladores, estes componentes são responsáveis por fornecer comunicação com o mundo exterior, podendo se dar de maneira digital e/ou analógica em alguns microcontroladores. Este componente é conhecido como GPIO, sigla para *General Purpose Input/Output*.

Os microcontroladores de forma geral trazem uma série de funcionalidades (JOHNS-[TON, 2018](#)) para os GPIO buscando torná-los versáteis e robustos a diversas aplicações.

Comparadores com histerese

Os comparadores com histerese, Schmitt-trigger, estão presentes em todas as portas GPIO de muitos microcontroladores e na maioria das vezes é utilizado com padrão ao utilizar um porta como entrada, seu papel é decidir se a entrada está em nível alto ou baixo. Ele conta dois limiares diferentes para a mudança de estado, o que torna a entrada mais imune a ruídos.

Resistores pull-up e pull-down

Em grande parte das aplicações não é interessante deixar uma entrada com uma tensão flutuante, a utilização de resistores de pull-up, conectando sinal à alimentação, e pull-down, sinal ao terra, é muito comum, e por isso alguns microcontroladores já trazem essa funcionalidade internamente.

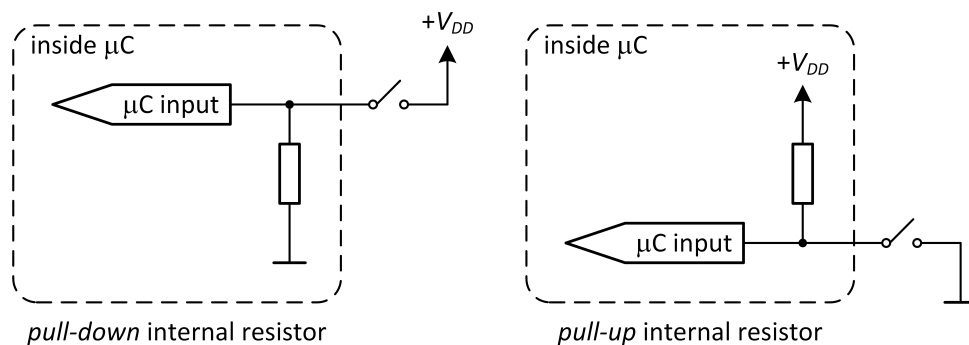


Figura 4 – Tipos de entrada

Fonte: ([ENABLED, 2019](#))

Deteção de borda

Além de ser possível identificar o nível lógico da entrada, grande parte dos microcontroladores também são capazes de detectar a variação desse sinal, nível lógico 0 para 1, borda de subida, e vice-versa, borda de descida.

Saída de coletor aberto, dreno aberto e push-pull

Os microcontroladores podem ter diversas formas de controlar sua saída, geralmente são utilizados transistores ou MOSFETs em dois tipos de configuração.

No caso da push-pull a topologia utilizada permite a saída ter dois valores de tensão, 0V ou VDD (figura 5), já para saídas do tipo coletor aberto (em transistores) ou dreno aberto (em MOSFETs) a saída pode assumir 0V ou alta impedância.

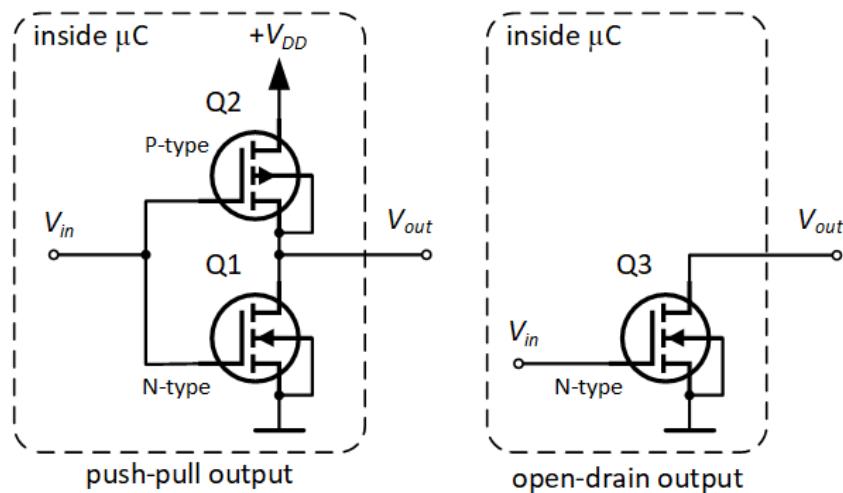


Figura 5 – Tipos de saída
 Fonte: (ENABLED, 2019)

2.1.1.4 Interrupções

Geralmente gerado por hardware, embora possa ser iniciado por software, as interrupções indicam que ocorreu um evento que precisa de uma resposta urgente. O valor de leitura de uma entrada é modificado, por exemplo, e é necessário verificar seu estado. O processador interrompe o que estava fazendo, armazena informações suficiente (*program counter* e *status register*) para que ele continue mais tarde e executa uma rotina de serviço de interrupção (ISR). Ele retorna à sua atividade anterior quando o ISR foi concluído. Assim, um ISR é algo como uma sub-rotina chamada por hardware e não por software (DAVIES, 2008).

2.1.1.5 Sinal de Relógio

Praticamente todos os sistemas digitais necessitam de um sinal de relógio para funcionarem, são conhecidos como pelo termo em inglês *clock*. É neste componente onde isso é gerado, pode ser obtido através de um cristal ou fontes externa. Atualmente os microcontroladores são capazes de gerar e trabalhar com uma faixa relativamente grande de frequências.

2.1.1.6 Temporizadores

Como uma extensão dos clocks, os temporizadores são muito versáteis e necessários para diversas funções do microcontrolador, como por exemplo gerar sinais de clock para outros periféricos do chip, calcular intervalos de tempo ou medir período de sinais.

Contadores

Um contador é um dispositivo que armazena o número de vezes que um evento ou processo específico ocorreu, geralmente em relação a um sinal de relógio. Os contadores são divididos em duas categorias, os síncronos e assíncronos e embora possam apresentar diversas topologias um circuito contador é geralmente construído com flip-flops conectados em cascata (FLOYD, 2007).

Watchdog Timer

Como ferramenta de segurança esse contador é responsável por resetar o microcontrolador caso o programa entre em um loop infinito.

2.1.1.7 Interfaces de comunicação

Ampliando a função de Entradas e Saídas, as interfaces de comunicação fornecem protocolos próprios para a transmissão de dados buscando sempre transmitir de maneira rápida e com baixas taxas de perda, dentre as mais comuns presentes nos microcontroladores temos como exemplo a Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C) e Universal Asynchronous Receiver/Transmitter (UART).

2.1.1.8 Conversores

Dados do mundo exterior são essencialmente analógicos, e para tornar a interação com circuitos que trabalham dessa maneira os conversores são peça indispensável. Essa conversão pode acontecer em dois sentidos, convertendo um sinal analógico em digital (A/D) e vice-versa (D/A).

Em ambos os casos existem duas grandezas de extrema importância, a resolução e a frequência de amostragem do sinal. A resolução está ligada diretamente ao erro de quantização e conseqüentemente à relação sinal-ruído da conversão. Já a frequência de amostragem está relacionada à velocidade com que o conversor pega amostras do sinal analógico.

Existem inúmeras topologias que tentam otimizar essas características de desempenho com simplicidade do circuito.

Comparadores

Utilizados em algumas topologias de conversores A/D, os comparadores são dispositivos que comparam duas tensões ou correntes e indicam qual delas é maior. Podem ser utilizados como detectores de passagem por zero e em alguns casos podem ser configurados para funcionarem como Schmitt trigger, no MAX971 ([INTEGRATED, 2003](#)) por exemplo.

2.1.2 Linguagens de Programação

A programação de um microcontrolador percorre diversos níveis de forma a converter o código escrito para a linguagem de máquina.

Linguagem de Máquina

A linguagem de máquina consiste em um código binário que o processador consegue entender. No início da computação a programação era feita dessa forma, mas hoje em dia não é mais necessário trabalhar nessa linguagem de baixo-nível.

Linguagem de Montagem

Um nível um pouco, só um pouco, acima da linguagem de máquina temos a linguagem de montagem, comumente conhecida como Assembly, ela traduz o código binário para algumas palavras conhecidas como mnemônicos.

Uma grande desvantagem é que essa tradução de código binário depende diretamente do processador, ou seja, cada tipo de processador tem um dicionário de palavras único. Em compensação ao utilizar o Assembly consegue-se ter mais controle sobre a eficiência do código e, além disso, é possível ver exatamente o comportamento de processador no processo de debug.

Linguagem C

Uma das linguagens de programação mais usadas para a programação de microcontroladores, por ser uma linguagem de alto-nível é possível criar lógicas complexas sem muito trabalho e sem a necessidade de conhecer todo conjunto de instruções do processador. A desvantagem fica por conta da compilação que pode não ser a mais eficiente, embora atualmente os compiladores estão cada vez melhores e trabalham muito bem nesse ponto.

2.1.3 Programando um Microcontrolador

Normalmente é possível programar um microcontrolador em qualquer uma das linguagens citadas acima, respeitando o conjunto de instruções de cada modelo.

Para efetivamente programar o chip normalmente é utilizado um ambiente de desenvolvimento integrado (IDE, do inglês Integrated Development Environment), existem diversos disponíveis, entretanto a maioria das fabricantes têm seu próprio ambiente como o Code Composer Studio da Texas Instruments e MPLAB X da Microchip.

No ambiente educacional existem também outras IDE que são muito utilizadas devido a sua maior simplicidade, como o Arduíno IDE e Energia, plataformas de código aberto com suporte à algumas famílias de microcontroladores.

2.1.4 Aplicações

Microcontroladores são utilizados em praticamente qualquer dispositivo automatizado, isso pode incluir sistemas de controle de automóvel, dispositivos médicos implantáveis, controles remoto, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos, entre muitos outros.

São diversos pontos que contribuem para sua grande utilização, mas o que mais pode-se destacar é seu custo e consumo de energia. Além disso, vantagens como facilidade de programação e arquitetura simplificada tornam os microcontroladores uma das melhores opções para prototipagem de projetos.

2.1.5 Uso da Educação

Se tratando especificamente de kits de desenvolvimento voltados para educação, trabalhos como [Martín-Ramos et al. \(2017\)](#), [Deaky, Lupulescu e Ursutiu \(2011\)](#), [Nunnally \(1996\)](#) e [Bittencourt da Silva e Patikowski Cheiran \(2015\)](#) conseguem destacar muito bem a importâncias deles no aprendizado em geral, não só no que tange ao conhecimento técnico. Embora este trabalho tenha como o foco a graduação, em [Silveira e Girardi \(2017\)](#) também pode se observar esse impacto no nível médio.

Ademais, existem alguns trabalhos que se posicionam de forma bem alinhada ao objetivo deste, a construção de kits e módulos próprios para auxiliar a aprendizagem no ambiente de graduação e afins como o caso de [Dabroom, Refie e Matmti \(2013\)](#) e [Su et al. \(2013\)](#).

2.2 Opções do Mercado

É muito importante mapear algumas opções de placas, módulos, componentes entre outros visando ter um olhar geral sobre o que é possível encontrar no mercado brasileiro, já que além de suprir as necessidades do aprendizado de microcontroladores, este trabalho também tem como objetivo construir shields acessíveis para os alunos.

2.2.1 Microcontroladores e Placas

Quando se trata do desenvolvimento de produtos, a utilização de microcontroladores não é feita usando somente o chip integrado, salvo em situações específicas, a desenvolvimento acontece sobre um kit, onde o microcontrolador é utilizado através de uma placa própria, onde a programação e o acesso ao pinos é muito mais fácil.

Kits de desenvolvimento já são muito comuns no mercado, são utilizados tanto no ensino quando em empresas para testes de novos produtos.

"[O Arduino, por exemplo,] se tornou um produto extremamente bem-sucedido junto a fabricantes, estudantes e artistas, devido à sua facilidade de uso e durabilidade" (MONK, 2013).

Outra placa que vem se destacando muito no mercado é a ESP32, com uma CPU de dois núcleos de 32 bits e integrada a módulos Wifi e Bluetooth é sem dúvida a placa com melhor custo-benefício do mercado.

Em contrapartida, temos a família de placas MSP430, com CPU de 16 bits apresentam um meio termo em as outras opções, mesmo não disponível no mercado brasileiro é muito utilizado no ambiente acadêmico e de empresas por se destacar muito no quesito consumo de energia.

Na tabela 1 encontra-se algumas das principais características das placas de desenvolvimento.

	CPU	RAM	Flash	GPIO	Preço
Arduino Uno	ATmega328 8-bit	2 KB	32 KB	20	~R\$ 50,00
Arduino Mega	ATmega2560 8-bit	8 KB	256 KB	70	~R\$ 90,00
MSP-EXP430G2ET	MSP430G2553 16-bit	512 B	16KB	16	\$ 10,00
MSP-EXP430F5529LP	MSP430F5529 16-bit	8 KB	128KB	32	\$ 13,00
ESP32	Xtensa Dual-Core 32-bit	520 KB	4 MB	25	~R\$ 70,00

Tabela 1 – Principais placas de desenvolvimento

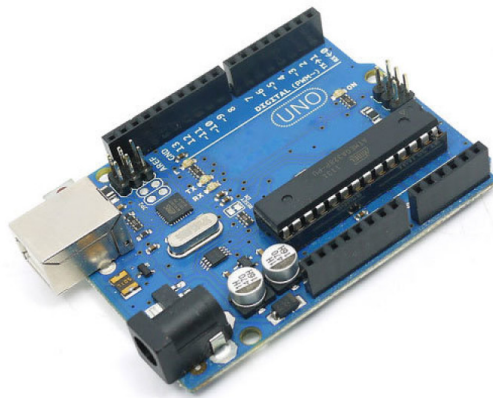
Todas as placas contam com suporte a comunicação serial e entradas analógicas com conversores A/D de 10 bits ou mais.

2.2.2 Shields

Para expandir as funcionalidades surgiram as placas conhecidas como shields, que por sua vez incorporam diferentes tipos de módulos para facilitar ainda mais o desenvolvimento e testes de protótipos.

No mercado brasileiro as shields ainda são restritas uma pequena variedade de módulos e, como maior impedimento, desenvolvidas para os kits com base em Arduino, modelos UNO e MEGA.

O uso desses shields é extremamente simples, bastando encaixá-los sobre o kit de desenvolvimento.



(a) Arduino Uno

Fonte: ([FILIPEFLOP, 2019a](#))



(b) Shield GSM

Fonte: ([MERCADOLIVRE, 2019](#))

Figura 6 – Kit e shield comum no mercado brasileiro.

2.2.2.1 BoosterPack

Os kits de desenvolvimento disponibilizados pela Texas Instruments têm como foco não só agilizar o desenvolvimento mas também auxiliar na educação, por isso além das LaunchPads a empresa conta com os *BoosterPack plug-in modules*.

Os BoosterPack são módulo que podem ser integrados aos kits de desenvolvimento para fornecer recursos e funcionalidades adicionais para fornecer recursos e funcionalidades adicionais, incluindo recursos de detecção de toque ([Texas Instruments, 2019d](#)), infravermelho ([Texas Instruments, 2019c](#)) e bateria ([Texas Instruments, 2019b](#)).

Além dos módulos já citados, a Texas Instruments também desenvolveu um BoosterPack com foco educacional, o *Educational BoosterPack MKII* ([Texas Instruments,](#)

2019a), ele conta com diversos módulos e um suporte especial através do *Energia*, ambiente de desenvolvimento de código aberto feito pela comunidade.

É possível encontrar outros kits que seguem a linha de BoosterPacks criados por grupos e/ou desenvolvedores (43OH, 2019) que serviram de inspiração para este trabalho já que todos os kits e módulos citados são extremamente escassos no Brasil, e seu custo de importação muitas vezes pode inviabilizar sua utilização.

2.2.3 Módulos e componentes

Como o objetivo deste trabalho é construir novas shields, é necessário conhecer opções de módulos e componentes disponíveis no mercado para tornar a etapa de fabricação viável.

2.2.3.1 Interfaces

Leds

Um dos componentes mais clássicos da eletrônica e utilizado em quase todos os projetos, também tem papel fundamental no aprendizado, sendo um dos primeiros componentes utilizados durante as aulas.

Basicamente podemos pensar em duas categorias básica, os LEDs monocromáticos e os RGB que podem ter sua cor controlada digitalmente.

Aqui características como consumo e tensão de operação são os pontos críticos. Na tabela 2 é possível observar algumas diferenças fundamentais.

	Tensão	Corrente	Dimensões
LED RGB	min. 3.2 V	30 mA	5 x 5 x 10 mm
LED Difuso	min. 2 V	20 mA	5 x 5 x 10 mm
LED Alto Brilho	min. 2 V	30 mA	5 x 5 x 10 mm

Tabela 2 – Comparação LEDs

Display

Displays de forma geral facilitam muito a visualização de informações, no mercado brasileiro existem algumas opções sendo 3 delas (figura 8) muito utilizadas em projetos.

	Tensão	Corrente	Tipo de Comunicação	Dimensões
Display OLED	3-5.5 V	25 mA	I2C	30 x 27 x 4 mm
Display Nokia 5110	3.3-5.5 V	50 mA	SPI	45 x 43 x 5 mm
Display 16x2	4.5-5.5 V	140 mA	Paralela 8 bits	80 x 36 x 12 mm

Tabela 3 – Comparação Displays

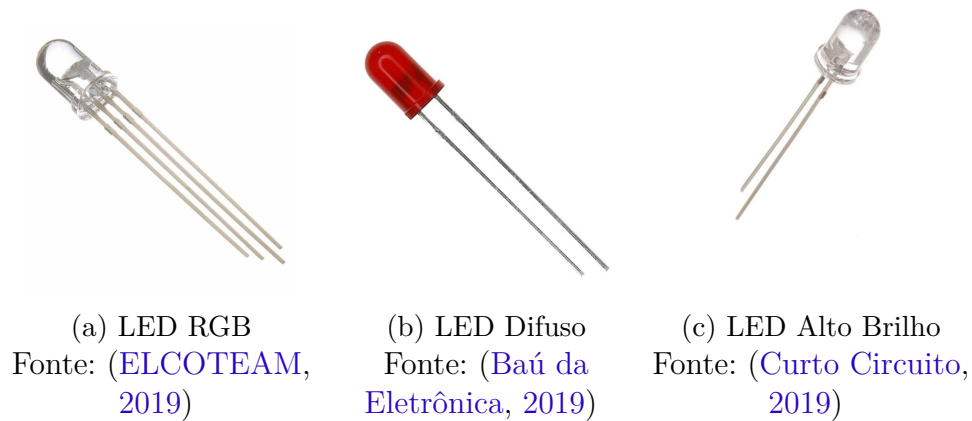


Figura 7 – Opções de LEDs

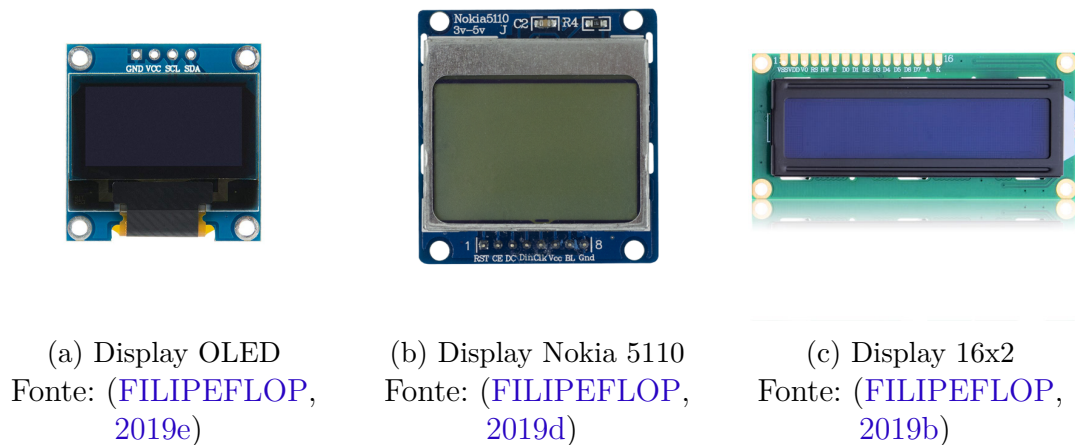


Figura 8 – Opções de displays

Display 7 Segmentos

Também na categoria de displays, o display de 7 segmentos serve com um meio termo entre os LEDs e Displays, sua utilização é muito vantajosa quando se trata de exibição de números, já que sua é muito mais simples do que um display comum e bem mais elegante do que a utilização de LEDs.

Essencialmente existe apenas um modelo de display de 7 segmentos, bastando utilizar componentes integrados extras para facilitar ainda mais sua utilização.

Apesar de versátil o display de 7 segmentos tem uma grande limitação que pode tornar sua utilização inviável, o número de conexões. Tal limitação pode ser resolvida com módulos um pouco mais complexos que utilizam da comunicação I2C por exemplo.

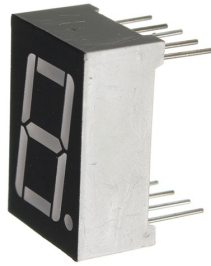


Figura 9 – Display 7 Segmentos

Fonte: (FILIPEFLOP, 2019c)

2.2.3.2 Sensores

Converter informação do ambiente em sinais elétricos é o princípio de funcionamento de um sensor, normalmente são a base de qualquer projeto.

Existem uma infinidade de sensores capazes de medir qualquer estímulo do ambiente, entretanto alguns deles apresentam certa versatilidade que pode ser muito útil para um kit de desenvolvimento.

Temperatura

Quando se pensa em sensores com certeza este é o primeiro a vir a mente, monitorar temperatura é só um exemplo da usabilidade desse sensor.

Novamente temos uma grande quantidade de modelos e aqui não temos muitas restrições, e por isso a característica mais importante para a escolha desse sensor deve ser sua forma de funcionamento, com ou sem contato.

Um dos modelos mais comuns no mercado é o DHT11 (AOSONG, 2010), que além de medir temperatura também conta com um sensor de umidade.

DTH11	
Tensão	3-5.5 V
Corrente	100-500 uA
Faixa de Temperatura	0-50 °C
Precisão de Temperatura	± 2 °C
Faixa de Umidade	20-90 %
Precisão de Umidade	± 5 %
Dimensões	18 x 12 x 5 mm

Tabela 4 – Características do Sensor de Temperatura e Umidade

Acelerômetro

O acelerômetro é outro sensor extremamente versátil, sua utilização é relacionada a qualquer tipo de projeto que trate de movimentação.

Existem basicamente 3 modelos disponíveis no mercado brasileiro o *MMA7361* ([Freescale Semiconductor, 2011](#)), o *ADXL335* ([Analog Devices, 2009](#)) e o *MPU-6050* ([INVENSENCE, 2013](#)), ambos com características semelhantes e comunicação serial, entretanto o último conta com um ponto de destaque, o mesmo circuito integrado contém um giroscópio e que o torna muito mais vantajoso.

MPU-6050	
Tensão	3-5 V
Conversor A/D	16 bits
Comunicação	I2C
Escalas Giroscópio	$\pm 250, 500, 1000, 2000$ °/s
Escalas Acelerômetro	$\pm 2, \pm 4, \pm 8, \pm 16$ g
Dimensões	2 x 1,6 x 0,1mm

Tabela 5 – Características do Acelerômetro

Luminosidade

Medir luminosidade é algo simples e restrito a um pequeno número de projetos, entretanto é algo de fácil implementação e favorece o entendimento sobre utilização do conversor A/D.

O sensor mais simples e que cumpre muito bem sua função é o LDR ([TECHNOLOGIES, 2008](#)), as opções de 5 e 3 mm devem ser consideradas.

Infravermelho

Como uma extensão dos sensores de luminosidade, o sensor infravermelho é capaz de detectar o espectro infravermelho de luz, na prática é uma luz invisível para o olho humano mas que é muito utilizado em controles remoto por exemplo.

Da mesma forma que o LDR, sua aplicação é limitada, mas se tratando de aprendizado, este módulo pode trazer uma nova perspectiva, já que seu funcionamento se baseia em pulsos de sinal.

O módulo encontrado no mercado é o *TSOP4838* ([SEMICONDUCTORS, 2018](#)).

Botões

Mais um dos componentes que são praticamente indispensáveis em qualquer projeto, os botões tem um funcionamento simples e são utilizados para interação do usuário

TSOP4838	
Tensão	2.7-5.5 V
Frequência	38 kHz
Ângulo de detecção	45°
Dimensões	30 x 5 mm

Tabela 6 – Características do Receptor Infravermelho

com o projeto.

Os botões e chaves de maneira geral podem ser classificados de acordo com duas características, número de circuitos que pode controlar ao mesmo tempo (polos) e quanto ao seu acionamento (único ou duplo).

Para o número de circuitos temos as siglas:

- SP, 1 polo;
- DP, 2 polos;
- 3P, 3 polos;
- 4P, 4 polos.

Com relação o tipo de funcionamento temos:

- ST para acionamento único;
- DT para acionamento duplo.

Além de seu funcionamento, o formato do botão também deve ser um fator relevante na escolha.

2.2.3.3 Atuadores

Atuadores é o nome dado a componentes que produzem movimento através de comandos elétrico e/ou mecânicos.

Relés

Comumente conhecido com relé, os interruptores eletromecânicos são, de maneira prática, componentes capazes de controlar eletronicamente circuitos de alta tensão.

Aqui temos poucas opções, principalmente porque seu funcionamento é extremamente simples, a grande diferença encontrada entre os modelos do mercado é com relação a sua tensão de ativação.



Figura 10 – Relé
Fonte: (FILIPEFLOP, 2019j)

Motores de Passo

A uso de motores de passo se faz com a utilização de controladores, conhecidos como drivers, capazes de fornecer a potência necessária para esses motores, é possível projetar esse tipo de circuito, entretanto isso foge do escopo deste trabalho, portanto neste projeto será utilizado módulos prontos capazes de executar tal função. Novamente existem diversas opções, mas temos uma grande restrição quando se trata desses drivers, o seu tamanho e dissipação de potência.

Para não esbarrar em nenhuma dessas restrições os módulos não serão de grande capacidade, controlando apenas motores de passo de pequeno porte. Considerando essa restrições temos as 3 (ST Microelectronics, 2002; Texas Instruments, 2014; MICROSYSTEMS, 2014) opções descritas na tabela 7

	Tensão de Alimentação	Corrente Máxima de Saída
ULN2003	5-12 V	500 mA
DRV8825	8.2-45 V	1.5 A
A4988	8-35 V	1 A

Tabela 7 – Comparação de drivers para motores de passo.

Motores de DC

Assim como para os motores de passo, os motores de corrente contínua (DC) também necessitam de um driver para fornecer potência, entretanto temos poucas opções disponíveis no mercado, sendo o L298N (ST Microelectronics, 2013) o mais comum.

L298N	
Tensão de Alimentação	7 - 35 V
Canais	2
Corrente máxima por canal	2 A

Tabela 8 – Especificação L298N

de comunicação tem se tornado cada vez mais comum na prototipagem de projetos, principalmente aqueles relacionados à Internet das Coisas (IoT).

RS232

Considerando a utilização de microcontroladores em ambientes industriais, o uso de protocolos mais robustos como o RS232 ([SEMICONDUCTOR, 1998](#)) é extremamente relevante, e para trazer esse ambiente mais próximo ao aluno, esse componente será levado em consideração na elaboração dos módulos.

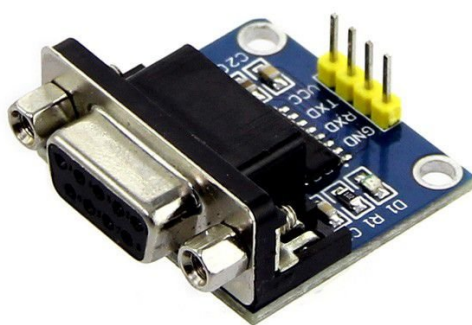


Figura 12 – Conversor RS232
Fonte: ([ELETROGATE, 2019b](#))

3 Desenvolvimento

Levando em conta todas as características e opções levantadas no capítulo anterior, é necessário analisar não só quais módulos devem compor as shields, mas principalmente como deve ser a organização deles para tornar sua utilização realmente relevante.

Antes de iniciar efetivamente com a escolha de módulos e sensores, será apresentada uma lista de requisitos que devem servir de referência em todas as escolhas, já que os requisitos descrevem todas as propriedades a serem atendidas pelo produto.

Além disso, como um dos objetivos deste trabalho, será necessário fazer a escolha do microcontrolador e do hardware que servirá de base para os módulos, da mesma forma para os componentes, esta etapa deverá atender os requisitos do projeto.

3.1 Requisitos

O projeto com um todo deve atender os seguintes requisitos:

- A placa de desenvolvimento base deve um modelo que já é utilizado em disciplinas de microcontroladores visando popularizar a utilização das shields desenvolvidas;
- Deve conter uma maneira clara de exibir dados, preferencialmente por meio de um display e uma interface mais simples de exibição;
- 3 ou mais botões para interação;
- 2 ou mais sensores ou módulos com comunicação I2C;
- 1 ou mais sensores analógicos;
- Driver para controle de 1 ou mais motores;
- 1 ou mais padrões de comunicação sem fio;
- As shields devem ter dimensões iguais ou muito próximas à da plataforma escolhida.

3.2 Restrições

A escolha dos componentes têm alguns restrições que afetam diretamente as opções disponíveis para o desenvolvimento deste trabalho. O tipo de montagem do componente é uma das limitações já que os testes do projeto utilizando componentes no formado SMD (*surface-mount device*) seriam muito mais complexos, além disso, como já citado, serão

selecionados apenas componentes e módulos disponíveis no mercado brasileiro visando reduzir custos de prototipagem.

3.3 Ferramentas

Até esta etapa foi utilizado apenas um software para elaboração dos esquemáticos e layouts, o *Eagle*.

Nas próximas etapas, o *Proteus* será o software responsável pela simulação de alguns subcircuitos. Além disso, para a programação do microcontrolador será utilizado o *Code Composer Studio* devido a escolha da placa de desenvolvimento, detalhada a seguir.

3.4 Proposta

A proposta em si consiste na elaboração de 4 shields que se adaptam entre si na Launchpad. Idealmente o usuário pode utilizar qualquer quantidade de shields diferentes e na ordem que preferir.

O principal cuidado que foi tomado durante o desenvolvimento foi com relação utilização adequada dos portas do microcontrolador, pois elas são relativamente limitadas. O anexo A foi utilizado como principal referência para seleção de portas a serem utilizadas pelos módulos.

3.4.1 Placa principal

Na FGA o microcontrolador utilizado é da Texas Instruments, a empresa conta com uma grande variedade de modelos com diferentes características e funcionalidades, entretanto buscando manter o modelo já utilizado na disciplina será utilizado a família de microcontroladores MSP430.

A Texas Instruments também disponibiliza um grande suporte para o desenvolvimento contando com datasheets detalhados e artigos sobre diversas funcionalidades de seus produtos, tudo pode ser encontrado em [Texas Instruments \(2019e\)](#).

3.4.2 MSP430

Essa linha de produtos tem como principal característica o baixo uso de energia, além disso este conjunto de microcontroladores conta suporte a periféricos digitais e analógicos integrados, o que facilita muito o desenvolvimento de projetos em geral.

3.4.2.1 LaunchPad MSP430

Launchpad foi o nome dado para o kit de desenvolvimento da Texas Instruments, se trata de uma placa de baixo custo que torna a utilização do microcontrolador muito mais simples, ele conta com pinos para acessar as conexões do chip além de um debugger que faz a conexão e programação do chip através de um cabo USB.

Entre as duas opções utilizadas na disciplina de Eletrônica Embarcada, a escolha foi o MSP430F5529 LaunchPad Evaluation Kit (figura 13) por contar com um maior número de portas disponíveis, característica necessária para atender todos os requisitos.



Figura 13 – LaunchPad Evaluation Kit.

As principais características desse modelo são:

- USB 2.0;
- Clock de até 25 MHz;
- 128KB de memória Flash e 8KB de RAM;
- Conversor A/D de 12 bits;
- Debugger open source integrado (eZ-FET lite);
- Entrada USB como fonte de 5V e 3.3V com conversor DC/DC de alta eficiência.

O número e tipo de portas também é outro fator relevante desta Launchpad, ela conta com 35 pinos de uso geral (GPIO), sendo:

- 8 com suporte à comparadores;

- 12 com suporte à temporizadores;
- 4 portas suportando 2 canais I2C;
- 9 portas suportando 3 canais SPI;
- 2 portas suportando UART.

Levando em conta que todos os tipos de comunicação serão utilizadas em apenas 1 canal, temos no total 28 portas disponíveis para a integração dos shields. As portas responsáveis pelos protocolos de comunicação estão representadas na tabela 10.

Porta	Periférico	
P3.3	UART	TX
P3.4		RX
P4.1	I2C	SDA
P4.1		SCL
P3.0	SPI	MOSI
P3.1		MISO
P3.2		CLK

Tabela 10 – Pinos de comunicação serial.

3.4.3 Informações

LEDs

A diferença de especificações entre os LEDs de alto brilho e difusos são mínimas, entretanto na prática a diferença de brilho é bem perceptível. Por conta desse detalhe a escolha será os LEDs de alto brilho.

O uso do LED RGB será feito em apenas uma das shields, a de Interface, já que é nela que deve se centralizar as informações.

Corrente de saída

A portas do microcontrolador só funcionam de forma estável com saída que utilizem até 15 mA ([Texas Instruments, 2018](#)), ou seja, para os LEDs e outros módulos que ultrapassem esse limite será necessária a utilização de um transistor.

Um dos modelos de transistor mais comum é o BC547, que por sua vez atende todas as necessidades desse projeto e portanto será o utilizado para fornecer a corrente não só para os LEDs, mas qualquer outro módulo.

Display

O grande fator limitante na escolha é o tamanho, pois apenas o display Oled [8a](#) é compatível com as dimensões da placa. Embora seja a única opção viável, esse display conseguirá atender praticamente todas as necessidades.

Display 7 segmentos

Como um intermediário, se tratando de complexidade de funcionamento, a shield também deve conter 4 display de 7 segmentos controlados por um conversor BCD-7segmentos (CD4511) e 4 transistores de ativação para o funcionamento por meio de multiplexação.

Botões

Por conta da limitação de espaço serão adicionados apenas 3 botões do tipo push-button SPST (figura [15e](#)) nesta shield, número suficiente para atender o requisitos.

Embora estes botões sejam os mais comuns, eles podem apresentar problemas com debounce, por isso e circuito RC simples foi adicionado ao esquemático visando minimizar esses erros.

3.4.3.1 Esquemático

As conexões devem seguir os dados da tabela [11](#).

Porta	Periférico		
P2.0	LED RGB	Vermelho	
P2.2		Verde	
P7.4		Azul	
P4.1	Display Oled	I2C	SDA
P4.2			SCL
P1.5	7 segmentos	Número	Bit 0
P1.4			Bit 1
P1.3			Bit 2
P1.2			Bit 3
P4.3		Multiplexação	Dígito 1
P4.0			Dígito 2
P3.7			Dígito 3
P8.2			Dígito 4
P7.0	Botões	BT0	
P3.6		BT1	
P3.5		BT2	

Tabela 11 – Conexões shield de interface

O circuito completo está representado no anexo [B](#).

3.4.3.2 Modelo PCB

A organização dos componentes selecionados devem seguir o modelo da figura 14.

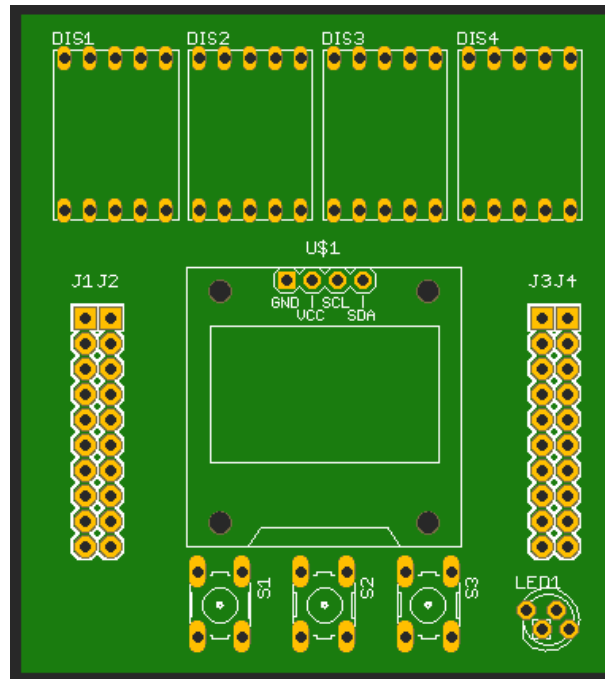


Figura 14 – Modelo da shield de interface
Fonte: Autoria própria

3.4.4 Sensores

Temperatura

De forma a tornar o projeto mais genérico, o modelo sem contado será o escolhido. Dentre as opções existentes no mercado brasileiro a mais comum e acessível é o DTH11 (figura 15a), além disso seu funcionamento é baseado em comunicação *one-wire*, podendo ser uma ótima maneira de explorar as funções do microcontrolador para implantar uma maneira de decodificar os dados recebidos. Vale ressaltar que além de temperatura este sensor também fornece valores de umidade, tornando-o ainda mais versátil.

Acelerômetro

Devida às vantagens já mencionadas na seção 2.2.3.2, a escolha para este projeto será o MPU 6050 (figura 8a).

Sensor de Luminosidade

Devida a sua simplicidade, o LDR (figura 15c) também deve compor a shield.

Sensor Infravermelho

Não tão simples, e com aplicações bem específicas, o TSOP4838 (figura 15d) deve integrar o shield para ter um possibilidade extra na elaboração de projetos.

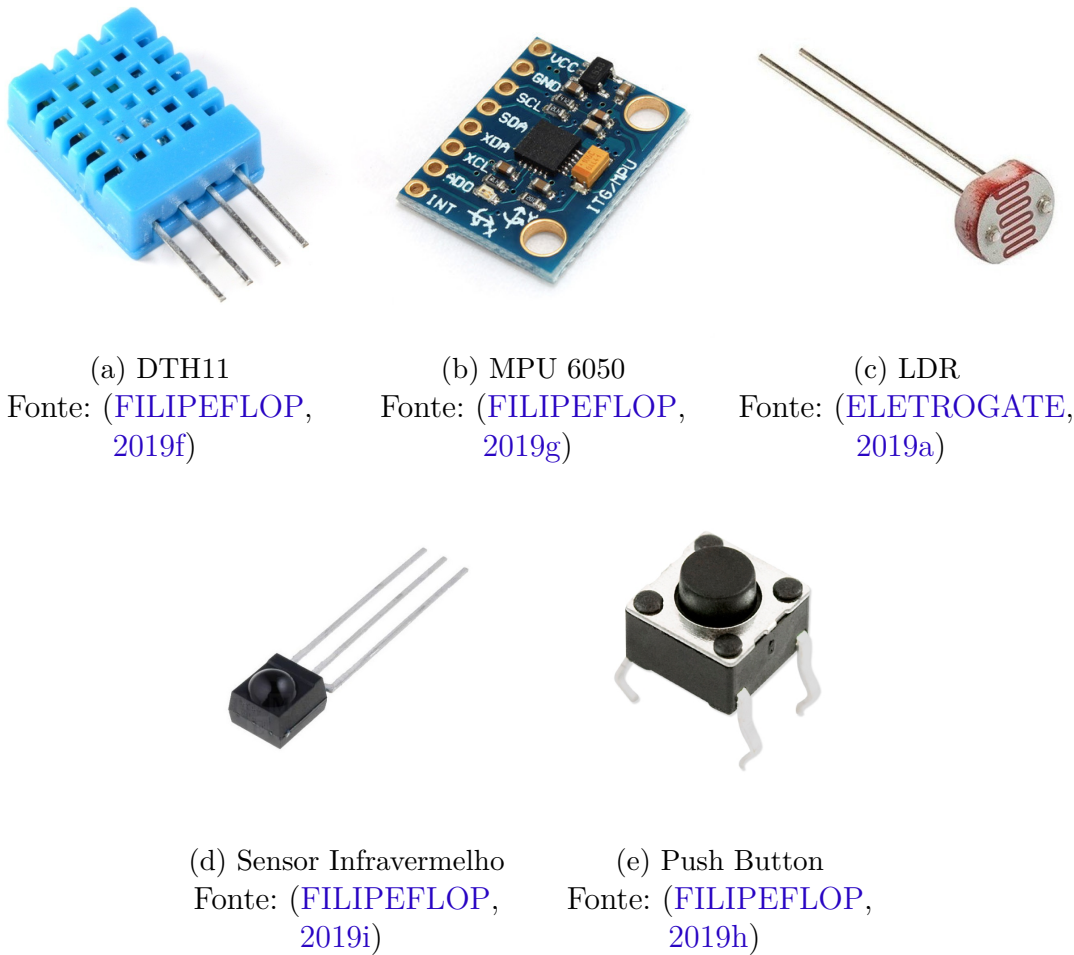


Figura 15 – Sensores

3.4.4.1 Esquemático

As conexões devem seguir os dados da tabela 12.

Porta	Periférico	
P6.5	LDR	
P1.6	Infravermelho	
P6.6	Temperatura	
P2.7	Acelerômetro	Interrupção
P4.2		SCL
P4.1		SDA

Tabela 12 – Conexões shield de sensores.

O circuito completo está representado no anexo B.

3.4.4.2 Modelo PCB

A organização dos componentes selecionados devem seguir o modelo da figura 16.

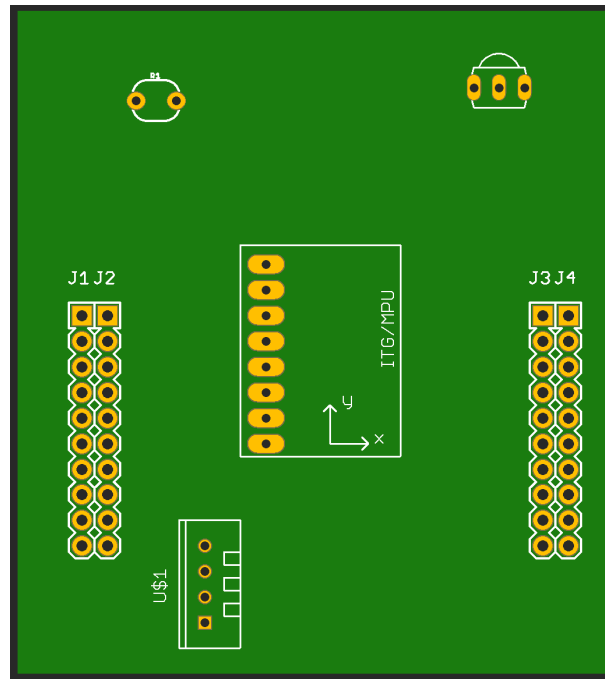


Figura 16 – Modelo da shield de sensores

Fonte: Autoria própria

3.4.5 Atuadores

Relé

A escolha do relé se dá basicamente por conta de sua tensão de operação, 5V no caso deste projeto.

Para seu correto funcionamento e segurança de utilização, é necessária a utilização de um circuito simples de proteção.

Driver para Motores

Conforme mencionado em 2.2.3.3, por conseguir atender o funcionamento de dois tipos de motores e ter boas especificações, o TB6612FNG se torna uma ótima opção para este projeto.

3.4.5.1 Esquemático

As conexões devem seguir os dados da tabela 13.

Portas	Periférico	
P6.0	Driver TB6612FNG	INA1
P6.1		INA2
P6.2		INB1
P6.3		INB2
P2.0		PWM A
P2.2		PWM B
P3.5	Relé	

Tabela 13 – Conexões shield atuadores.

3.4.6 Conectividade

Por atender perfeitamente os requisitos, a ESP32 será a escolha para ter as conexões Wifi e Bluetooth, entretanto como já temos uma placa de desenvolvimento predefinida faz-se necessário apenas a funcionalidade de conectividade sem fio que o ESP32 fornece, portanto neste projeto será utilizado apenas o *core* do kit, o ESP32-WROOM-32, figura 17



Figura 17 – ESP32 Wroom 32

Fonte: ([INDIAMART](#), 2019)

Para a utilização deste módulo será necessário um circuito auxiliar para a programação do chip ([SYSTEMS](#), 2019). Além disso será utilizado um firmware específico para o funcionamento por meio de comandos AT ([SYSTEMS](#), 2018).

As conexões necessárias para a comunicação com a Launchpad estão na tabela 14.

Porta	Periférico	
P3.4	ESP32-wroom-32	UART RX
P3.3		UART TX
P3.5		Enable

Tabela 14 – Conexões shield de conectividade.

3.5 Cronograma

As próximas etapas devem seguir o descrito na tabela 15.

Atividade	Jan	Fev	Mar	Abr	Mai	Jun	Jul
Compra dos módulos e sensores	■						
Simulação dos subcircuitos		■					
Elaboração dos layouts		■					
Teste dos módulos e sensores		■					
Elaboração do firmware		■	■				
Elaboração da documentação		■	■	■			
Correções nos layouts				■	■		
Pedido das PCIs				■		■	
Testes da placas					■		■
Documentação dos resultados		■	■	■		■	■
Revisão do documento							■
Elaboração da apresentação						■	
Apresentação							■

Tabela 15 – Cronograma.

4 Conclusão

Durante a elaboração deste trabalho foi possível destacar as dificuldades encontradas por alunos e desenvolvedores durante a construção de um projeto, seja ele um projeto pessoal ou algo desenvolvido para uma disciplina da graduação por exemplo.

Como primeira etapa, este trabalho visou dar uma introdução ao funcionamento dos microcontroladores, mas principalmente mostrar como é o mercado brasileiro de módulos e componentes utilizados em prototipagem, mostrando como as shields podem ajudar no processo como um todo.

Desenvolver shields voltadas para microcontroladores se mostra uma ótima forma de ajudar o desenvolvimento de projetos e consequentemente o aprendizado na área. Utilizando como base a Launchpad MSP430F5529 este projeto pode se tornar extremamente atrativo para os alunos de Engenharia Eletrônica da UnB.

Para finalizar este projeto ainda existem algumas etapas importantes que ainda não foram totalmente exploradas neste documento, principalmente quando se trata da integração dos componentes tanto em hardware quanto em software. Além disso este projeto exige um certo período de testes e algumas validações para realmente torná-lo viável.

Referências

43OH. *BoosterPacks*. 2019. Disponível em: <<http://store.43oh.com/BoosterPacks>>. Citado na página 21.

Analog Devices. *Datasheet: ADXL335*. 2009. Disponível em: <<https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>>. Citado na página 24.

AOSONG. *Datasheet: DHT11*. 2010. Disponível em: <http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001405544-da-01-en-TEMP_UND_FEUCHTESENSOR_DHT11.pdf>. Citado na página 23.

Baú da Eletrônica. *LED Difuso*. 2019. Disponível em: <<https://www.baudaeletronica.com.br/led-difuso-5mm-vermelho.html>>. Citado na página 22.

Bittencourt da Silva, V.; Patikowski Cheiran, J. Análise do uso de microcontroladores como ferramenta de apoio ao ensino-aprendizagem de Arquitetura de Computadores. *International Journal of Computer Architecture Education*, 2015. Citado na página 18.

Curto Circuito. *LED Alto Brilho*. 2019. Disponível em: <<https://www.curtocircuito.com.br/led-alto-brilho-vermelho-5mm.html>>. Citado na página 22.

DABROOM, A.; REFIE, W. M.; MATMTI, R. Microcontroller-based learning kit course design using constructive alignment principles. *2013 21st Mediterranean Conference on Control and Automation, MED 2013 - Conference Proceedings*, IEEE, p. 558–566, 2013. Citado na página 18.

DAVIES, J. *MSP430 Microcontroller Basics*. [S.l.: s.n.], 2008. ISBN 9780750682763. Citado 3 vezes nas páginas 11, 13 e 15.

DEAKY, B.; LUPULESCU, N. B.; URSUTIU, D. Extended educational use of the Microcontroller Student Learning Kit (MCU SLK). *2011 IEEE Global Engineering Education Conference, EDUCON 2011*, IEEE, p. 913–916, 2011. Citado na página 18.

Dinis Gaspar, P.; Espírito Santo, A.; RIBEIRO, B. MSP430 microcontrollers essentials - A new approach for the embedded systems courses: Part 1 - Overview and Tools. *4th European Education and Research Conference (EDERC 2010)*, IEEE, p. 66–70, 2010. Citado na página 11.

Dinis Gaspar, P.; Espírito Santo, A.; RIBEIRO, B. MSP430 microcontrollers essentials - A new approach for the embedded systems courses: Part 2 - System and Peripherals. *4th European Education and Research Conference (EDERC 2010)*, IEEE, p. 76–80, 2010. Citado na página 11.

Dinis Gaspar, P.; Espírito Santo, A.; RIBEIRO, B. MSP430 microcontrollers essentials - A new approach for the embedded systems courses: Part 3 - Data Acquisition and Communications. *4th European Education and Research Conference (EDERC 2010)*, p. 205–209, 2010. Citado na página 11.

- ELCOTEAM. *LED RGB*. 2019. Disponível em: <https://www.elcoteam.com/aziende_en/diodo-led-rgb-5mm-tricolore-catodo-comune-pth-ll-509rgbm2e-002.html>. Citado na página 22.
- ELETROGATE. *LDR*. 2019. Disponível em: <<https://www.eletrogate.com/sensor-fotoresistor-ldr-de-5mm>>. Citado na página 35.
- ELETROGATE. *RS232*. 2019. Disponível em: <<https://www.eletrogate.com/conversor-serial-rs232-ttl-max3232>>. Citado na página 28.
- ENABLED, M. *Digital inputs and outputs*. 2019. Disponível em: <<https://os.mbed.com/teams/TVZ-Mechatronics-Team/wiki/Digital-inputs-and-outputs>>. Citado 2 vezes nas páginas 14 e 15.
- FILIPEFLOP. *Arduino Uno Rev3*. 2019. Disponível em: <<https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/>>. Citado na página 20.
- FILIPEFLOP. *Display 16x2*. 2019. Disponível em: <<https://www.filipeflop.com/produto/display-lcd-16x2-backlight-azul/>>. Citado na página 22.
- FILIPEFLOP. *Display 7 Segmentos*. 2019. Disponível em: <<https://www.filipeflop.com/produto/display-7-segmentos-1-digito-vermelho/>>. Citado na página 23.
- FILIPEFLOP. *Display Nokia 5110*. 2019. Disponível em: <<https://www.filipeflop.com/produto/display-lcd-nokia-5110-backlight-azul/>>. Citado na página 22.
- FILIPEFLOP. *Display Oled*. 2019. Disponível em: <<https://www.filipeflop.com/produto/display-oled-0-96-polegadas-i2c-branco/>>. Citado na página 22.
- FILIPEFLOP. *DTH11*. 2019. Disponível em: <<https://www.filipeflop.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>>. Citado na página 35.
- FILIPEFLOP. *MPU 6050*. 2019. Disponível em: <<https://www.filipeflop.com/produto/acelerometro-e-giroscopio-3-eixos-6-dof-mpu-6050/>>. Citado na página 35.
- FILIPEFLOP. *Push Button*. 2019. Disponível em: <<https://www.filipeflop.com/produto/chave-tactil-push-button-x10-unidades>>. Citado na página 35.
- FILIPEFLOP. *Receptor Infravermelho*. 2019. Disponível em: <<https://www.filipeflop.com/produto/receptor-infravermelho-ir-tsop4838/>>. Citado na página 35.
- FILIPEFLOP. *Relé 5V*. 2019. Disponível em: <<https://www.filipeflop.com/produto/rele-5v-single-srd-05vdc-sl-c/>>. Citado na página 26.
- FLOYD, T. *Sistemas Digitais: Fundamentos e Aplicações*. 9a ed. ed. Porto Alegre, RS: Bookman Editora LTDA, 2007. v. 53. ISSN 1098-6596. Citado na página 16.
- Freescall Semiconductor. *Datasheet: MMA7361LC*. 2011. Citado na página 24.
- HARARI, E.; NORMAN, R. D.; MEHROTRA, S. *Flash eeprom system*. [S.l.]: Google Patents, 1994. US Patent 5,297,148. Citado na página 12.
- INDIAMART. *ESP32 Wroom*. 2019. Disponível em: <<https://www.indiamart.com/proddetail/esp32-wroom-32-20563569548.html>>. Citado na página 37.

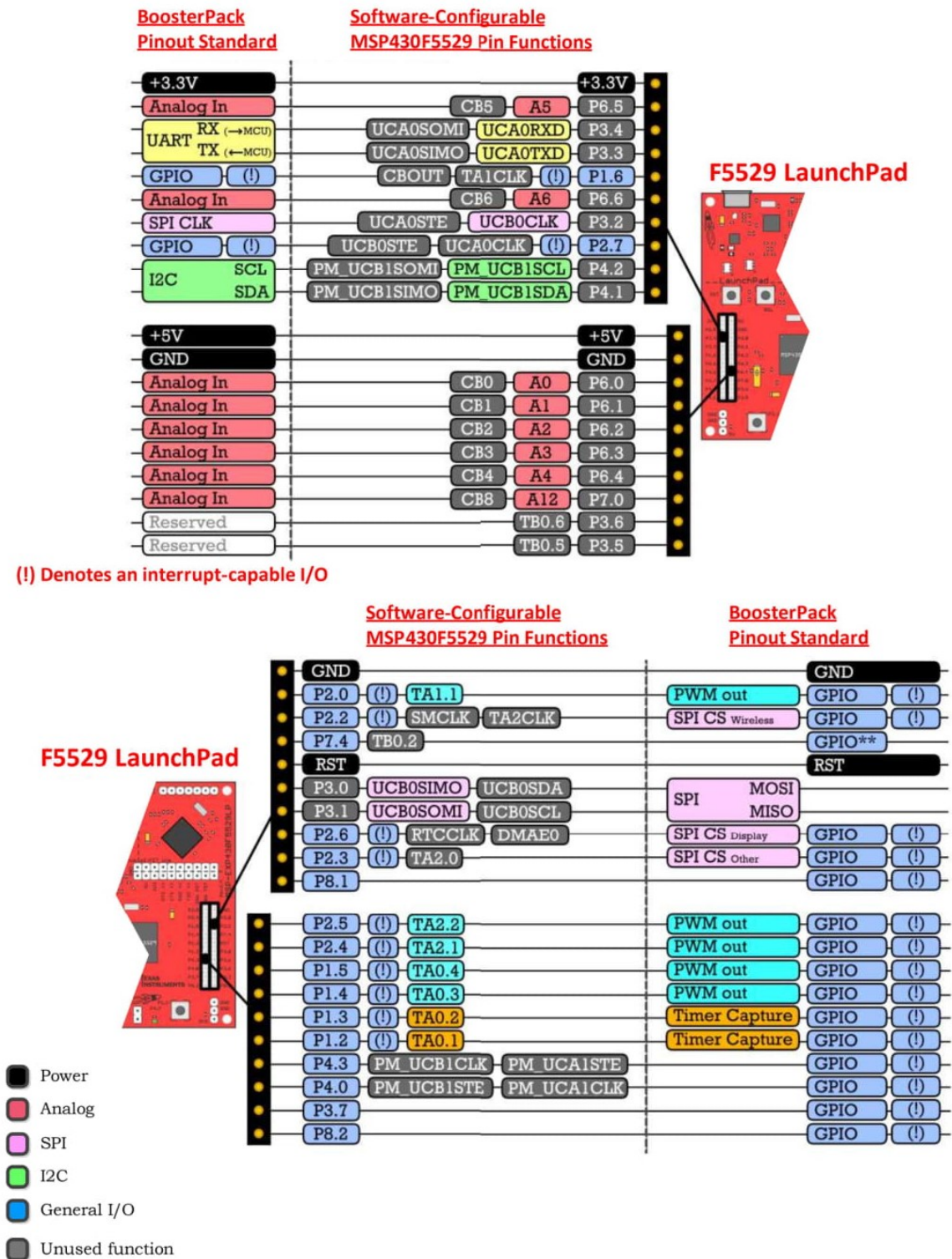
- INTEGRATED, M. *Datasheet: MAX971*. 2003. Citado na página 17.
- INVENSENCE. *Datasheet: MPU-60x0*. 2013. Disponível em: <<https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>>. Citado na página 24.
- JOHNSTON, P. *Demystifying Microcontroller GPIO Settings*. 2018. Disponível em: <<https://embeddedartistry.com/blog/2018/06/04/demystifying-microcontroller-gpio-settings/>>. Citado na página 14.
- LINEBACK, R. Microcontrollers Will Regain Growth After 2019 Slump. *IC Insights*, 2019. Citado na página 8.
- MARTÍN-RAMOS, P. et al. First exposure to Arduino through peer-coaching: Impact on students' attitudes towards programming. *Computers in Human Behavior*, v. 76, p. 51–58, 2017. ISSN 07475632. Citado na página 18.
- MERCADOLIVRE. *Shield GSM*. 2019. Disponível em: <<http://bit.ly/344OYiO>>. Citado na página 20.
- MICROSYSTEMS, A. *Datasheet: A4988*. 2014. Disponível em: <https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf>. Citado na página 26.
- MONK, S. *Programação com Arduino: Começando com sketches*. Porto Alegre, RS: Bookman Editora LTDA, 2013. Citado na página 19.
- NUNNALLY, C. E. Teaching microcontrollers. *Proceedings - Frontiers in Education Conference*, p. 434–436, 1996. ISSN 01905848. Citado na página 18.
- OLIVEIRA, N. *Paralelo entre as arquiteturas von Neumann e Harvard*. 2014. Citado na página 13.
- SEMICONDUCTOR, D. Fundamentals of RS-232 Serial Communications. 1998. Citado na página 28.
- SEMICONDUCTORS, V. *Datasheet: TSOP48*. 2018. Disponível em: <<https://www.vishay.com/docs/82459/tsop48.pdf>>. Citado na página 24.
- SILVEIRA, S.; GIRARDI, M. Desenvolvimento de um kit experimental com Arduino para o ensino de Física Moderna no Ensino Médio. *Revista Brasileira de Ensino de Física*, v. 39, n. 4, 2017. ISSN 01024744. Citado na página 18.
- SPARKFUN. *TB6612FNG*. 2019. Disponível em: <<https://www.sparkfun.com/products/14450>>. Citado na página 27.
- ST Microelectronics. *Datasheet ULN200xA*. 2002. Disponível em: <<https://www.seeedstudio.com/document/pdf/ULN2003Datasheet.pdf>>. Citado na página 26.
- ST Microelectronics. *Datasheet: L298N*. 2013. Disponível em: <https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf>. Citado na página 26.
- SU, S. et al. Teaching embedded programming to electrical engineers, BioEngineers, and mechanical engineers via the escape platform. *Proceedings of the 3rd Interdisciplinary Engineering Design Education Conference, IEDEC 2013*, IEEE, p. 87–92, 2013. Citado na página 18.

- SYSTEMS, E. *ESP32 AT Instruction Set and Examples*. 2018. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_at_instruction_set_and_examples_en.pdf>. Citado na página 37.
- SYSTEMS, E. *Datasheet: ESP32-WROOM-32*. 2019. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>. Citado na página 37.
- TECHNOLOGIES, S. *Datasheet: LDR*. 2008. Disponível em: <<https://www.sunrom.com/get/443700>>. Citado na página 24.
- Texas Instruments. *MSP430F5529 LaunchPad Development Kit*. 2017. Citado na página 45.
- Texas Instruments. *Datasheet: MSP430F552x , MSP430F551x Mixed-Signal Microcontrollers*. 2018. Citado na página 32.
- Texas Instruments. *Educational BoosterPack MKII*. 2019. Disponível em: <<http://www.ti.com/tool/BOOSTXL-EDUMKII>>. Citado na página 21.
- Texas Instruments. *Fuel Tank MKII Battery BoosterPack Plug-In Module*. 2019. Disponível em: <<http://www.ti.com/tool/BOOSTXL-BATPAKMKII>>. Citado na página 20.
- Texas Instruments. *Infrared (IR) BoosterPack Plug-in Module*. 2019. Disponível em: <<http://www.ti.com/tool/BOOST-IR>>. Citado na página 20.
- Texas Instruments. *MSP430 CapTIvate™ Touch Keypad BoosterPack*. 2019. Disponível em: <<http://www.ti.com/tool/BOOSTXL-CAPKEYPAD>>. Citado na página 20.
- Texas Instruments. *MSP430 ultra-low-power MCUs – Design & development*. 2019. Disponível em: <<http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/design-development.html>>. Citado na página 30.
- Texas Instruments. *Datasheet: DRV8818*. 214. Disponível em: <<http://www.ti.com/lit/ds/symlink/drv8825.pdf>>. Citado na página 26.
- TOSHIBA. *Datasheet: TB6612FNG*. 2008. Disponível em: <https://www.pololu.com/file/download/TB6612FNG.pdf?file_id=0J86>. Citado na página 27.

Anexos

ANEXO A – LaunchPad Development Kit

Pinout



Fonte: (Texas Instruments, 2017)

ANEXO B – Esquemáticos

