

PLATAFORMA DE PROGRAMAÇÃO VISUAL PARA ESP8266

Roberto Luiz Debarba

Prof. Miguel Alexandre Wisintainer – Orientador

1 INTRODUÇÃO

Atualmente, os estudantes do ensino básico estão imersos em um ambiente em que a tecnologia é facilmente percebida: carros, celulares e computadores são exemplos que todos conhecem e muitos utilizam, no entanto, poucos entendem. Estes mesmos estudantes passam boa parte de seu tempo na escola estudando conteúdos de matemática e física e, paradoxalmente, os conceitos que lhes são apresentados parecem distantes (BENITTI et al., 2009).

Von Wangenheim, Nunes e Santos (2014) afirmam que, apesar da computação estar presente em todos os setores da sociedade hoje, existe uma carência de conhecimento e interesse da população nesta área. Complementam informando que uma das razões é a ausência do ensino de computação no ensino fundamental.

Uma forma de viabilizar o conhecimento científico-tecnológico, segundo Benitti et al. (2009), e, ao mesmo tempo estimular a criatividade e a experimentação com um forte apelo lúdico, pode ser proporcionada através da robótica educacional. Hoje esse recurso ainda é pouco utilizado, principalmente no cenário nacional, porém, conforme afirma Sentance et al. (2017, tradução nossa), com os últimos avanços em tecnologia educacional, o preço dos recursos permitem aos educadores utilizarem ambientes de simulação baseados em paradigmas de ensino digital (*e-learning*), com baixo custo e que podem ser aplicados diretamente na sala de aula em larga escala.

Diante do exposto, este trabalho propõe o desenvolvimento de uma plataforma para programação visual do microcontrolador ESP8266, que permitirá, através de componentes de baixo custo, aplicar em larga escala a robótica educacional nas salas de aula.

1.1 OBJETIVOS

O objetivo deste trabalho é disponibilizar uma plataforma de programação visual baseada em blocos para suporte ao ensino de robótica educacional nas escolas.

Os objetivos específicos são:

- a) disponibilizar uma plataforma de programação visual baseada em blocos;
- b) permitir a execução dos programas desenvolvidos no microcontrolador ESP8266 (Espressif Systems);

- c) garantir a execução dos programas de forma direta, sem a necessidade de o usuário interagir com programas terceiros.
- d) apresentar ao usuário o código fonte gerado para consulta e edição.

2 TRABALHOS CORRELATOS

São apresentados três trabalhos correlatos, que possuem características e objetivos semelhantes ao tema proposto. O primeiro trabalho, descrito na seção 2.1, não está diretamente relacionado ao objetivo de estudo do tema. Nele é apresentado um estudo de caso da aplicação da ferramenta Scratch no ensino fundamental (VON WANGENHEIM; NUNES; SANTOS, 2014). A seção 2.2 descreve uma plataforma de robótica educacional para programação em Scratch chamada de Scratchboard (VON WANGENHEIM et al., 2017). Por fim, a seção 2.3 apresenta uma plataforma de robótica educacional com programação visual em blocos, que inclui hardware e software próprios, conhecida como Micro:bit (PROTZENKO, 2015).

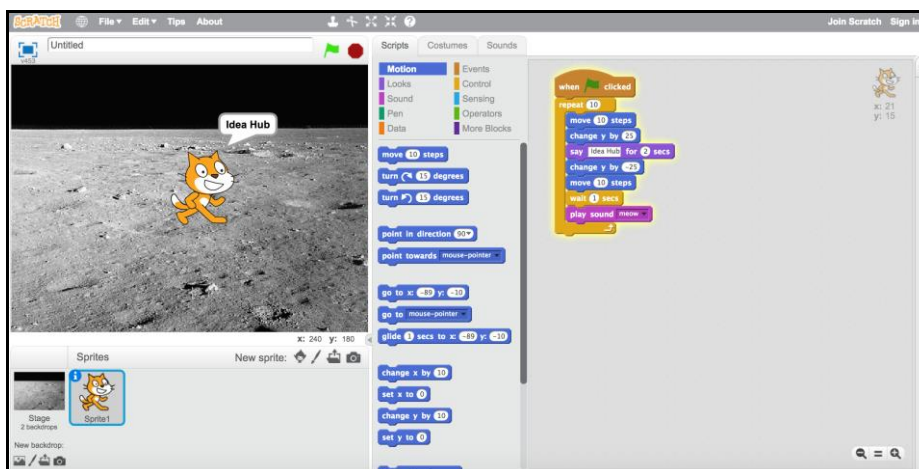
2.1 ESTUDO DE CASO DA APLICAÇÃO DA FERRAMENTA SCRATCH

Von Wangenheim, Nunes e Santos (2014) desenvolveram um estudo de caso que visa aplicar e avaliar uma estratégia para o ensino de computação e programação nas escolas, focando no ensino fundamental. A pesquisa tem como principal questionamento a viabilidade de desenvolver uma unidade instrucional interdisciplinar usando o Scratch, cujo ambiente de desenvolvimento por ser visto na Figura 1, para ensinar computação de forma adequada.

A ferramenta Scratch permite a programação visual, por blocos, de estruturas básicas, variáveis, condições, repetições, entradas e saídas de dados e abstrações pelo uso de funções. Não é possível visualizar o código programado ou gerado em modo texto, ou executar o programa fora da ferramenta ou em hardwares não suportados, como microcontroladores. O Scratch possui material de apoio em português do Brasil e pode ser baixada gratuitamente no site oficial.

No estudo, as seguintes características foram analisadas: quais conceitos de programação e recursos do ambiente são aprendidos pelos alunos no final da unidade instrucional; se os objetivos de aprendizagem são atingidos usando a unidade instrucional; se Scratch facilita a aprendizagem da prática computacional/programação; se o Scratch motiva os alunos a aprender computação; se o Scratch promove uma experiência de usuário agradável e divertida.

Figura 1 - Ambiente de desenvolvimento do programa Scratch



Fonte: elaborado pelo autor.

Segundo Von Wangenheim, Nunes e Santos (2014), no âmbito do contexto identificado foi desenvolvida uma unidade instrucional para o ensino de conceitos de computação, para ser ao mesmo tempo inspiradora e envolvente, ajudando os alunos a encarar a computação como uma parte importante do seu mundo. Focou-se na aprendizagem ativa, criatividade, exploração incorporada em outras áreas curriculares, como Literatura e Artes.

Concluindo, Von Wangenheim, Nunes e Santos (2014) demonstram que o estudo indica o sucesso do ensino de computação usando Scratch já no primeiro ano do Ensino Fundamental. Os alunos avaliados conseguiram programar uma história interativa em poucas aulas. Também conclui que o ensino de computação pode ser integrado no currículo existente de forma harmônica e interdisciplinar. Observou-se que as aulas motivaram os alunos a aprender mais sobre programação e promoveram uma experiência de aprendizagem positiva e satisfatória.

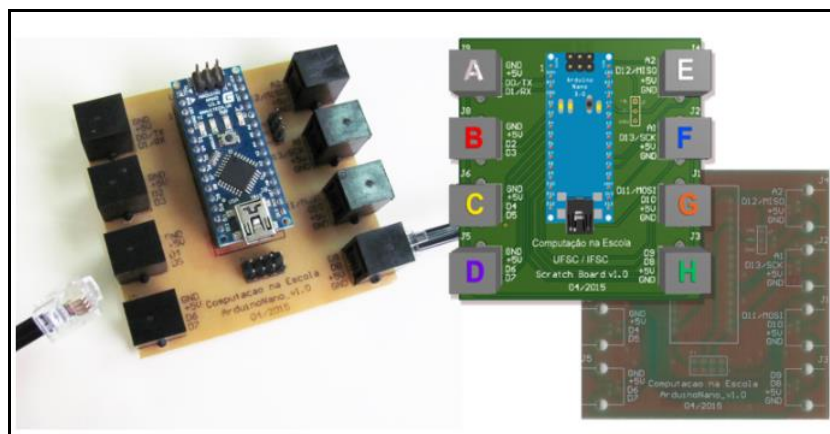
2.2 SCRATCHBOARD

Von Wangenheim et al. (2017) apresentam o desenvolvimento de uma plataforma de baixo custo para ensino de programação composta pelo microcontrolador Arduino Nano (VON WANGENHEIM et al., 2017), componentes de hardware, uma extensão para Scratch, plano de ensino e material de apoio. A Figura 2 apresenta uma foto real e uma representação gráfica do hardware do projeto, onde pode ser identificado o Arduino Nano ao centro e os conectores de componentes nas laterais.

O microcontrolador Arduino Nano permite a inclusão de sensores e atuadores compatíveis com Arduino, que podem ser adquiridos em qualquer fornecedor. Como a

programação é realizada no Scratch, que é distribuído gratuitamente pelo desenvolvedor oficial, não existe custo com software. O desenvolvimento em Scratch também agrega um formato de programação visual por blocos. O autor não deixa claro se o acesso ao código fonte gerado é possível, mas com a interpretação geral do trabalho entende-se que a funcionalidade não está disponível.

Figura 2 - Foto real e representação gráfica do Scratchboard



Fonte: Von Wangenheim et al. (2017).

O estudo apresenta a aplicação da plataforma através de um *workshop* para professores do ensino médio que tem como objetivo ensinar a programar um robô super-herói. Os assuntos abordados são comandos e eventos simples, condicionais, repetições, operadores, interfaces de leitura e escrita para componentes externos e como criar e usar funções.

Como conclusão, Von Wangenheim et al. (2017) afirmam que os resultados preliminares indicam que o *workshop* e a plataforma desenvolvidos são efetivos e motivaram a maioria dos participantes a introduzir computação em suas aulas. O *workshop* também resultou em professores com entendimento básico de computação e programação, possibilitando a integração da plataforma de forma multidisciplinar em suas aulas. Von Wangenheim et al. (2017) também destaca que a aplicação da tecnologia pelos professores em sala de aula requer treinamentos mais longos para que possam prover o suporte necessário.

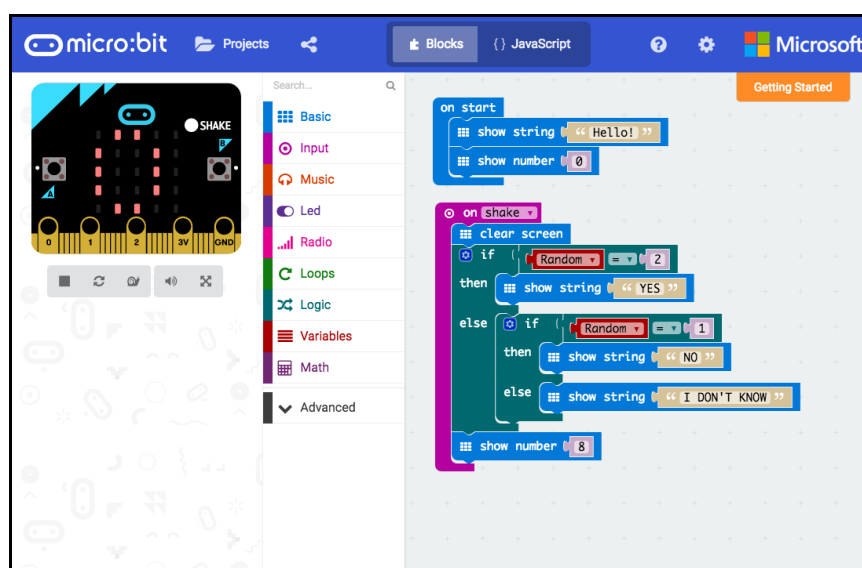
2.3 MICRO:BIT

Desenvolvido por Protzenko (2015), o projeto Micro:bit é uma plataforma que tem por objetivo original o ensino de programação para crianças de 11 a 12 anos de idade no Reino Unido. Apesar da idade e região especificadas originalmente, hoje o projeto abrange diversos países, incluindo o Brasil, e aplica-se a estudantes das mais diversas idades. Com sua

utilização é esperado que os estudantes compreendam estruturas de código simples, condicionais, repetições e abstrações via funções.

A plataforma é composta por duas partes: hardware e aplicação de programação. O hardware conta com um microcontrolador e uma série de sensores e atuadores embutidos. São eles: um compasso, um acelerômetro, bluetooth de baixa energia (BLE), uma série de pinos de entrada e saída, um sistema de LEDs 5x5 e dois botões. Com os pinos de entrada e saída é possível adicionar sensores e atuadores de terceiros. O ambiente de desenvolvimento, mostrado na Figura 3, é visual, baseada em blocos e construída com a biblioteca Blockly (PROTZENKO, 2015). Não necessita instalação e é executada diretamente do navegador, mas requer acesso à internet. A instalação da ferramenta de forma offline não é suportada, tornando-a inacessível aos locais sem acesso à internet. A plataforma está disponível em português, porém seu material de apoio oficial não foi traduzido para o idioma.

Figura 3 - Ambiente de desenvolvimento do programa Micro:bit



Fonte: Protzenko (2015).

3 PROPOSTA DA PLATAFORMA

Este capítulo tem como objetivo apresentar a justificativa para o desenvolvimento deste trabalho, os principais requisitos e a metodologia de desenvolvimento que será utilizada. Também são relacionadas às fontes bibliográficas que irão fundamentar o estudo proposto.

3.1 JUSTIFICATIVA

O Quadro 1 apresenta de forma comparativa as características das plataformas descritas nos trabalhos correlatos. Pode-se observar que as três plataformas disponibilizam o formato de programação visual, através de blocos, que é sempre destacada como sua principal

característica. Porém a única ferramenta que também permite a programação por linguagem textual é a Micro:bit, permitindo que usuários mais avançados continuem seus estudos sem a necessidade de comprar outros equipamentos.

Referente à programação de microcontroladores, exceto pelo Scratch, todas as plataformas possuem suporte. O Scratchboard possibilita a compra do hardware de outros fornecedores e a montagem do projeto de forma independente. No Micro:bit a única forma de programar o hardware é adquirindo-o através dos fornecedores oficiais. No caso do Scratch, a programação de microcontroladores não é suportada, pois o objetivo é o desenvolvimento de aplicações visuais que rodam diretamente na ferramenta. Enquanto as demais ferramentas são orientadas à sensores e atuadores físicos, o Scratch utiliza a interação entre objetos virtuais e entrada de dados e ações por mouse e teclado. Por não precisar de nenhum hardware adicional, o Scratch é mais acessível, porém pode despertar menos o interesse dos alunos. As duas plataformas que suportam a programação de microcontroladores permitem a extensão com componentes de terceiros.

Ao avaliar o quesito acessibilidade no Brasil, as diferenças entre as ferramentas tornam-se maiores. As três estão disponíveis oficialmente no Brasil, com a venda por fornecedores oficiais daquelas que dependem de hardware. Quanto à documentação, tanto o Scratch quanto o Scratchboard possui material em português do Brasil. O Micro:bit não possui documentação traduzida para o idioma.

Outro quesito importante para acessibilidade no Brasil, principalmente ao considerar as escolas públicas, é o acesso à internet. Para atingirmos o maior número de alunos, a possibilidade de instalação e execução offline é fundamental. O Scratch e o Scratchboard permitem a execução offline, sendo o primeiro também disponível via navegador de internet. A plataforma Micro:bit é acessível apenas através da internet.

Por fim, considerando que as plataformas necessitam de componentes de hardware para execução, o preço é uma característica importante. As três ferramentas estão disponíveis gratuitamente para acesso e/ou *download* no site oficial. Como o Scratch não possui hardware, o custo de aquisição para o usuário é zero, tornando-se a ferramenta mais acessível. No caso do Scratchboard, o hardware não é vendido oficialmente pelo projeto, sendo necessário buscar seus componentes em lojas não relacionadas ao projeto. O valor do equipamento base, o Arduino Nano, é estimado em R\$35,00, porém apenas com ele não é possível interagir com o mundo físico, sendo necessária a compra de atuadores e sensores. O custo final varia de acordo com os valores dos componentes escolhidos. Finalmente, a ferramenta Micro:bit possui revendedor oficial no Brasil e seu hardware possui os

componentes necessário para a elaboração de projetos. Seu valor sugerido é de R\$150,00.

Quadro 1 - Comparativo entre os trabalhos correlatos

trabalhos características	Scratch (VON WANGENHEIM; NUNES; SANTOS, 2014)	Scratchboard (VON WANGENHEIM et al., 2017)	Micro:bit (PROTZENKO, 2015).
programação visual	X	X	X
programação por texto			X
programação de hardware		X	X
hardware extensível		X	X
material de apoio em português	X	X	
pode ser instalado	X	X	
pode ser executado pelo navegador de internet	X	X	X
disponível oficialmente no Brasil	X	X	X
valor estimado no Brasil	R\$ 0,00	mais de R\$35,00	R\$150,00

Fonte: elaborador pelo autor.

A partir das características apresentadas acima, conclui-se que a ferramenta mais acessível é o Scratch, por ser totalmente gratuita e não possuir hardware. Apesar do seu baixo custo, tende a despertar menos o interessante dos alunos e não permite a aplicação robótica educacional. As outras plataformas possuem suporte à robótica educacional e formato de programação similar ao Scratch, porém seu custo de aquisição é elevado, impedindo o acesso em comunidades de baixa renda.

Assim propõe-se o desenvolvimento de uma plataforma de programação visual baseada em blocos para a aplicação de robótica educacional, com suporte ao microcontrolador ESP8266. Nessa plataforma também será possível visualizar e editar o código fonte gerado com base nas aplicações desenvolvidas, permitindo a utilização para estudos mais avançados. Com essa solução, tem-se a intenção de possibilitar a o ensino de programação em comunidades de baixa renda, usando a robótica educacional como uma ferramenta para

aumentar o interesse dos alunos pela área e aumentando as possibilidades de aplicação multidisciplinar. No campo tecnológico o trabalho torna-se relevante ao propor o estudo da criação de uma linguagem de programação visual para hardware com a biblioteca Blocky e a geração de código executável para o microcontrolador ESP8266, utilizando técnicas de implementação de compiladores.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A plataforma desenvolvida deve:

- a) possibilitar o desenvolvimento de programas através de uma linguagem de programação visual baseada em blocos e uma linguagem textual (Requisito Funcional – RF);
- b) gerar o código de execução do programa desenvolvido para o microcontrolador suportado (RF);
- c) permitir ao usuário visualizar o código gerado para o microcontrolador a partir da linguagem de programação visual (RF);
- d) executar o programa desenvolvido diretamente no microcontrolador sem a necessidade de o usuário interagir com ferramentas de terceiros (RF);
- e) suportar como microcontrolador o ESP8266 da Espressif Systems (Requisito Não Funcional – RNF);
- f) utilizar a biblioteca Blocky como plataforma para a implementação da linguagem de programação visual baseada em blocos (RNF);
- g) codificada com a linguagem de programação TypeScript (RNF);
- h) gerar o código para execução no microcontrolador suportado no dialeto da plataforma Arduino (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: realizar o levantamento bibliográfico sobre o ensino de computação e programação nas escolas, as ferramentas de programação visual e robótica educacional já disponíveis e trabalhos correlatos;
- b) elicitação de requisitos: detalhar e reavaliar os requisitos e, se necessário, especificar novos e remover existentes a partir das necessidades observadas durante o desenvolvimento do estudo;

- c) especificação e análise: formalizar a linguagem de programação visual através da notação Backus-Naur Form (BNF) e formalizar as funcionalidades da ferramenta através dos diagramas de classe e de atividades da Unified Modeling Language (UML), utilizando a ferramenta Star UML.
- d) implementação: implementar a plataforma proposta a partir dos requisitos definidos;
- e) testes: paralelamente à implementação, elaborar e executar testes com o objetivo de validar se os comandos criados na linguagem de programação visual são suficientes para a elaboração de programas variados e se o código para o microcontrolador suportado está sendo gerado corretamente.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2019									
	jul.		ago.		set.		out.		nov.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
especificação e análise										
implementação										
testes										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Este capítulo está dividido em três seções. A seção 4.1 aborda uma visão geral sobre o conceito de linguagem de programação visual. A seção 2.3 descreve as características e abordagens da robótica educacional, enquanto a seção 4.3 apresenta os recursos e especificações do ESP8266.

4.1 LINGUAGEM DE PROGRAMAÇÃO VISUAL

Segundo Jost et al. (2014, tradução nossa), uma linguagem de programação visual (VPL) é uma linguagem de programação que permite ao usuário criar programas principalmente através de manipulações gráficas. Fraser (2015) aponta que linguagens de programação visuais baseadas em blocos são normalmente um ponto de início para a programação. No contexto de ensino de programação, é um caminho que vicia os alunos no assunto, antes de movê-los a conteúdos mais complexos.

Aprender com uma linguagem de programação textual pode ser realmente estressante [...]. VPLs proveem vantagens para iniciantes comparadas às linguagens baseadas em texto,

como uma melhor experiência de usuário, maior facilidade ao aprender uma nova linguagem e ainda previnem erros típicos de sintaxe [...]. (JOST et al., 2014, tradução nossa)

Hoje, Arduino é uma linguagem de programação de robôs amplamente utilizada. Segundo Booth e Stumpf (2013, tradução nossa), apesar da linguagem Arduino ser mais fácil de aprender que muitas outras linguagens de programação, ela ainda necessita o domínio de diversas habilidades para potenciais usuários finais. Recentemente, novas VPLs para Arduino têm sido desenvolvidas e a facilidade de iniciação em programação provida pela representação visual dessas linguagens impressiona. Isso se relaciona com os objetivos de outras VPLs que estão surgindo, como, por exemplo, motivar crianças e facilitar o processo de aprendizagem. (BOOTH; STUMPF, 2013, tradução nossa)

4.2 ROBÓTICA EDUCACIONAL

Machado (2017) define a robótica educacional como um ambiente de aprendizagem que utiliza a tecnologia como instrumento no qual os participantes podem vivenciar experiências semelhantes ao cotidiano.

Segundo Queiroz, Sampaio e Santos (2016), a robótica educacional permite ao estudante manipular e controlar objetos concretos e, através destes, observar a materialização dos comandos dados por eles ao computador, processo a partir do qual se estabelece a construção do seu conhecimento.

No ambiente exposto, para Maissonnette (2002), o aluno é levado a observar, abstrair e inventar, criando modelos a partir de peças de brinquedos ou eletrodomésticos danificados, de peças de kits Lego ou Arduino e circuitos eletrônicos.

Neste contexto, é importante tornar o aluno um agente ativo de conhecimento, possibilitando-o construir todo o seu processo de aprendizagem. Defende-se a inclusão da robótica educacional como ferramenta desta transformação, já que o aluno tem a responsabilidade da busca do entendimento necessário para que seu projeto alcance o objetivo proposto (CASTILHO, 2009).

A robótica educacional interfere de forma significativa na motivação dos alunos, uma vez que ensina de forma prática a aplicação da tecnologia em problemas corriqueiros (FORNAZA; WEBBER, 2014). O ensino de robótica educacional permite a elaboração de um currículo interdisciplinar, visto que para solucionar um problema o aluno deve ser capaz de relacionar diversos conhecimentos e competências, como por exemplo, ler, articular e interpretar símbolos ou código em diferentes linguagens (SANTOS et al., 2010; SCHIVANI, 2014).

4.3 ESP8266

ESP8266 é um microcontrolador desenvolvido pela Espressif Systems, cuja fabricação e comercialização iniciaram em 2014. Uma das características do microcontrolador é o suporte à conexão Wireless Fidelity (Wi-Fi), possibilitando ao dispositivo atuar como Access Point (AP), estação ou ambos. O ESP8266 é fornecido em vários modelos, sendo a principal diferença entre eles a quantidade de entradas e saídas genéricas (*General-purpose input/output* – GPIO) disponíveis (KOLBAN, 2015).

O microcontrolador é comercializado em diversos modelos, chamados módulos, cuja variação está principalmente nos tipos de conexões e quantidade de GPIOs.

Todos os modelos do ESP8266 podem servir como uma ponte serial Wi-Fi, ou seja, um Arduino pode-se conectar a uma rede Wi-Fi através da comunicação serial com o ESP8266 (CURVELLO, 2015).

Quadro 3 – Especificação do ESP8266

Voltage	3.3V
Current consumption	10uA – 170mA
Flash memory attachable	16MB max (512K normal)
Processor	Tensilica L106 32 bit
Processor speed	80-160MHz
RAM	32K + 80K
GPIOs	17 (multiplexed with other functions)
Analog to Digital	1 input with 1024 step (10 bit) resolution
802.11 support	b/g/n/d/e/i/k/r
Maximum concurrent TCP connections	5

Fonte: Kolban (2015).

O processador do ESP8266 é de 32-bits e possui uma frequência de 80MHz, podendo operar em até 160MHz. A tensão de alimentação do ESP8266 é de 3,3 volts. As características completas são apresentadas no Quadro 3. O firmware padrão do ESP8266 vem com um interpretador de comandos AT, para realizar a comunicação com redes Wi-Fi. A ferramenta Arduino IDE possui integração com o ESP8266 para gravar o firmware e o

programa escrito na linguagem C++ através da comunicação serial, porém, também existem firmwares para interpretação de outras linguagens de programação como: Lua, Javascript e Python (ACROBOTIC, 2016, tradução nossa).

REFERÊNCIAS

ACROBOTIC. **Getting Started With The ESP8266 ESP-12E Development Board**. Disponível em: <<http://learn.acrobotic.com/tutorials/post/esp8266-getting-started>>. Acesso em: 11 set. 2016.

BENITTI, Fabiane Barreto Vavassori et al. Experimentação com Robótica Educativa no Ensino Medio: ambiente, atividades e resultados. **Anais do Workshop de Informática na Escola**. Blumenau, Sc, p. 1811-1820. jan. 2009. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/2166>>. Acesso em: 26 set. 2018.

BOOTH, Tracey; STUMPF, Simone. End-User Experiences of Visual and Textual Programming Environments for Arduino. **End-user Development**, [s.l.], p.25-39, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-38706-7_4.

CASTILHO, Maria I. **Robótica na educação: com que objetivos?** [Porto Alegre], [2009]. Disponível em: <<http://www.pucrs.br/eventos/desafio/2007/mariaines.php#raclog>>. Acesso em: 15 set. 2016.

CURVELLO, André. **Apresentando o módulo ESP8266**. 2015. Disponível em: <<http://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 25 out. 2016.

FORNAZA, Roseli; WEBBER, Carine G. Robótica educacional aplicada à aprendizagem em física. **Renote**, [Porto Alegre], v. 12, n. 1, p. 1-10, jul. 2014. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/50275/31405>>. Acesso em: 02 nov. 2016.

FRASER, Neil. Ten things we've learned from Blockly. **2015 Ieee Blocks And Beyond Workshop (blocks And Beyond)**, [s.l.], p.49-50, out. 2015. IEEE. <http://dx.doi.org/10.1109/blocks.2015.7369000>.

JOST, Beate et al. Graphical Programming Environments for Educational Robots: Open Roberta - Yet Another One?. **2014 Ieee International Symposium On Multimedia**, [s.l.], p.381-386, dez. 2014. IEEE. <http://dx.doi.org/10.1109/ism.2014.24>.

KOLBAN, Neil. **Kolban's book on the ESP8266**. 2016. 436 p. Disponível em: <<http://neilkolban.com/tech/esp8266/>>. Acesso em: 18 nov. 2018.

MACHADO, João Paulo. **ROBOTÓY**: Aplicação para programação e simulação de robôs. 2017. TCC (Graduação) - Curso de Ciência da Computação, Fundação Universidade Regional de Blumenau, Blumenau, 2017.

MAISONNETTE, Rogers. **A utilização dos recursos informatizados a partir de uma relação inventiva com a máquina**: a robótica educativa. [Paraná], [2002]. Disponível em: <<http://www.proinfo.gov.br/upload/biblioteca.cgd/192.pdf>>. Acesso em: 02 nov. 2016.

PROTZENKO, Jonathan. Pushing blocks all the way to C++. **2015 Ieee Blocks And Beyond Workshop (blocks And Beyond)**, [s.l.], out. 2015. IEEE. <http://dx.doi.org/10.1109/blocks.2015.7369010>.

SANTOS, Franklin L. et al. REDUC: a robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 16., 2010, Santos. **Anais eletrônicos...** [S.l.]: SBC, 2010. P. 1304-1313. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/2053>>. Acesso em: 05 nov. 2016.

SCHIVANI, Milton. **Contextualização no ensino de física à luz da teoria antropológica do didático: o caso da robótica educacional**. 2014. 220 f. Tese (Doutorado em Educação) – Faculdade de Educação, Universidade de São Paulo, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/48/48134/tde-01122014-104322/>>. Acesso em: 05 nov. 2016.

SENTANCE, Sue et al. "Creating Cool Stuff": Pupils' Experience of the BBC micro:bit. **Proceedings Of The 2017 Acm Sigcse Technical Symposium On Computer Science Education - Sigcse '17**, [s.l.], p.1-2, 2017. ACM Press. <http://dx.doi.org/10.1145/3017680.3017749>.

QUEIROZ, Rubens; SAMPAIO, Fábio Ferrentini; SANTOS, Mônica Pereira dos. DuinoBlocks4Kids: Ensinando conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional. **Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (cbie 2016)**, [s.l.], p.1169-1178, 10 nov. 2016. Sociedade Brasileira de Computação - SBC. <http://dx.doi.org/10.5753/cbie.wcbie.2016.1169>.

VON WANGENHEIM, Aldo et al. Motivating Teachers to Teach Computing in Middle School – A Case Study of a Physical Computing Taster Workshop for K-12 Teachers. **International Journal Of Computer Science Education In Schools**, [s.l.], v. 1, n. 4, p.35-50, 31 out. 2017. ICT in Practice. <http://dx.doi.org/10.21585/ijcses.v1i4.17>.

VON WANGENHEIM, Christiane Gresse; NUNES, Vinícius Rodrigues; SANTOS, Giovane Daniel dos. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. **Revista Brasileira de Informática na Educação**, [s.l.], v. 22, n. 03, p.115-125, 23 nov. 2014. Comissão Especial de Informática na Educação.
<http://dx.doi.org/10.5753/rbie.2014.22.03.115>.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO (PROJETO) – PROFESSOR TCC I

Acadêmico(a): _____

Avaliador(a): _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
ASPECTOS METODOLÓGICOS	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			
	9. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?			
	10. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?			
	11. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?			
	As citações obedecem às normas da ABNT?			
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?			

PARECER – PROFESSOR DE TCC I OU COORDENADOR DE TCC:

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 4 (quatro) itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos 4 (quatro) itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: () APROVADO () REPROVADO

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

FORMULÁRIO DE AVALIAÇÃO (PROJETO) – PROFESSOR AVALIADOR

Acadêmico(a): _____

Avaliador(a): _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			

PARECER – PROFESSOR AVALIADOR:

O projeto de TCC será reprovado, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **5 (cinco)** tiverem resposta ATENDE PARCIALMENTE.

PARECER: () APROVADO () REPROVADO

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.