

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO
AULAS GALILEU

Luiz Fernando de Andrade Gadêlha

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Mecatrônico*

Banca Examinadora

Prof. Alexandre Zaghetto, CIC/UnB <i>Orientador</i>	_____
Prof. Fulano de Tal 2, ENE/UnB <i>Co-orientador</i>	_____
Prof. Fulano de Tal 3, EESC/USP <i>Examinador externo</i>	_____
Prof. Fulano de Tal 4, ENE/UnB <i>Examinador interno</i>	_____

Dedicatória

Dedico este trabalho em primeiro lugar a Deus, por todas benções que me fizeram continuar e todas dificuldades que me fizeram crescer. Dedico este trabalho também à minha família, que esteve comigo em todos momentos da minha formação.

Luiz Fernando de Andrade Gadêlha

Agradecimentos

Agradeço a meus colegas de curso, projetos e ao meu professor orientador por este trabalho

Luiz Fernando de Andrade Gadêlha

RESUMO

Este trabalho tem como objetivo propor um curso prático em Algoritmos e Programação de Computadores Utilizando a placa Intel® Galileo voltada para alunos de graduação dos curso de engenharia mecatrônica, elétrica e de computação. Tal proposta se fundamenta na noção de que a inclusão de práticas laboratoriais . A disciplina tem como base de desenvolvimento o microcontrolador Galileo e conceitos de eletrônica de todos níveis.

ABSTRACT

This work aims to propose a discipline aimed at undergraduate students for learning development of embedded circuits. This proposal is based on the growing need and popularization of embedded circuits geared to various purposes, such as home automation, building and industrial. The course has the development of basic microcontroller Galileo and electronics concepts of all levels.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	OBJETIVO	2
1.2	APRESENTAÇÃO DO MANUSCRITO	3
2	REVISÃO BIBLIOGRÁFICA	4
2.1	PLACA INTEL [®] GALILEO	4
2.1.1	PINAGEM DA PLACA INTEL [®] GALILEO	4
2.1.2	CARACTERÍSTICAS ELÉTRICAS E ELETRÔNICAS DA PLACA INTEL [®] GALILEO	5
3	DESENVOLVIMENTO	9
3.1	INTRODUÇÃO	9
3.2	ARQUITETURA GERAL	9
3.3	CLASSIFICADOR ESTATÍSTICO DE PADRÕES	9
3.4	SEÇÃO	10
3.4.1	SUB-SEÇÃO	10
4	RESULTADOS EXPERIMENTAIS	11
4.1	INTRODUÇÃO	11
4.2	AVALIAÇÃO DO ALGORITMO DE RESOLUÇÃO DA EQUAÇÃO ALGÉBRICA DE RICCATI	11
5	CONCLUSÕES	13
	REFERÊNCIAS BIBLIOGRÁFICAS	14
	ANEXOS	15
I	DIAGRAMAS ESQUEMÁTICOS	16
II	DESCRIÇÃO DO CONTEÚDO DO CD	17

LISTA DE FIGURAS

2.1	Descrição dos pinos da placa Galileo - parte frontal[1]	5
2.2	Descrição dos pinos da placa Galileo - parte traseira[1]	5
2.3	Sinal PWM.....	7

LISTA DE TABELAS

2.1	Pinos Galileo[1].....	6
4.1	Tempos de execução em segundos para diferentes máquinas	11

LISTA DE SÍMBOLOS

Símbolos Latinos

A	Área	$[m^2]$
C_p	Calor específico a pressão constante	$[kJ/kg.K]$
h	Entalpia específica	$[kJ/kg]$
\dot{m}	Vazão mássica	$[kg/s]$
T	Temperatura	$[^{\circ}C]$
U	Coefficiente global de transferência de calor	$[W/m^2.K]$

Símbolos Gregos

α	Difusividade térmica	$[m^2/s]$
Δ	Variação entre duas grandezas similares	
ρ	Densidade	$[m^3/kg]$

Grupos Adimensionais

Nu	Número de Nusselt
Re	Número de Reynolds

Subscritos

amb	ambiente
ext	externo
in	entrada
ex	saída

Sobrescritos

\cdot	Variação temporal
$—$	Valor médio

Siglas

ABNT Associação Brasileira de Normas Técnicas

Capítulo 1

Introdução

A economia mundial está passando por uma grande revolução neste século. As bases econômicas de muitos países, outrora baseadas em *extração de matérias-primas* e *indústrias de transformação* de tais matérias primas, são agora baseadas em *conhecimento e transmissão de informação* [2]. O desenvolvimento tecnológico da Ciência da Computação é a principal responsável por tal revolução e todas Engenharias e Ciências Exatas são, direta ou indiretamente, influenciados por ela. Neste contexto, são fundamentais os conhecimentos e habilidades relacionadas a Ciência da Computação para o desenvolvimento de todas as Engenharias. Em especial, é importante sua forma de ensino e aprendizagem para a realidade na qual vivemos atualmente.[3].

A educação como a conhecemos atualmente foi idealizada na Prússia, no final do século XVII. Tal modelo educacional é chamada de *aprendizagem centrada no professor* ou *aprendizagem passiva*). No modelo de *aprendizagem passiva*, os estudantes são meros receptores do conhecimento oriundo do professor. Tal modelo se adequou bem as necessidades econômicas da época. Nessa época, era exigido do trabalhador habilidades ligadas a pura repetição e obediência[4].

Hoje em dia, principalmente por causa da revolução engredada pela computação, boa parte das universidades no mundo já começaram a modificar seus paradigmas educacionais realizando uma transição da *aprendizagem passiva*) para o modelo de *aprendizagem ativa*) ou *aprendizagem centrada no aluno*). Nesse modelo, o estudante é o principal responsável por sua aprendizagem e o professor é o orientador das experiências de ensino. Com esse modelo, tem-se conseguido obter altos índices de paradigmas ligados a criatividade, liderança, trabalho em equipe, gerenciamento e auto-gerenciamento além de um aumento substancial na motivação dos estudantes, visto que eles podem se apropriar verdadeiramente de seu processo de aprendizagem além terem se mostrado próprios para a aprendizagem e ensino de conceitos ligados a computação[5].

A realidade da educação brasileira, no entanto, não tem acompanhado as tendências supracitadas. Técnicas eficientes de ensino de conhecimentos relacionados a ensino de Ciências Exatas e Engenharia são parte estratégica para o desenvolvimento de qualquer país. Entretanto, as mudanças nos cursos de engenharia, no Brasil, em geral têm sido relacionadas a simples adição ou supressão de conteúdos,mas não uma revisão profunda das bases de ensino, levando em consideração as transformações atuais[6]. Segundo o jornal *A Gazeta do Povo*, a taxa de evasão no curso de

engenharia no Brasil é de aproximadamente 57% [7] e em geral a evasão ocorre nas partes iniciais dos cursos, onde os alunos têm seu primeiro contato com computação . Pode-se afirmar, tendo em vista essa estatística, que o paradigma educacional atual é o maior responsável pelo grande déficit de engenheiros qualificados no Brasil.

Com relação ao ensino de habilidades e conceitos de computação básica, no mundo se observa - não apenas no Brasil - que os estudantes usualmente têm grandes dificuldades de aprendizagem, que o conhecimento e aprendizagem dos alunos tende a se estagnar nos níveis mais rasos de entendimento, de forma que os conhecimentos não são interconectados, mas apenas específicos ao contexto estudado[8] além de, muitas vezes, se sentirem desmotivados devido a fragmentação do conhecimento nas disciplinas[6, 5]. Tais problemas de aprendizagem também se mostram presentes nos profissionais que saem das faculdades. Boa parte dos profissionais, no Brasil, possuem formação deficiente. Não tem capacidade plenamente desenvolvida para serem *aprendizes-estudantes autônomos*. Tal habilidade é essencial para terem sucesso na economia mundial atual, que é centrada em conhecimento e informação[3].

Para realizar a transição entre o modelo *aprendizagem-passiva* para o modelo *aprendizagem-ativa* , muitas universidades já se utilizam de placas eletrônicas com microcontroladores como Arduino e similares[9]. Nas referências citadas, o ensino de computação básica aliada a projetos práticos tem alcançado grande aumento nos índices acadêmicos dos alunos e diminuição nas taxas de evasão.

Este trabalho tem como objetivo primário a proposta de um curso prático para a disciplina *Algoritmos e Programação de Computadores* utilizando a placa de desenvolvimento Intel® Galileo com dinâmicas pedagógicas próprias do paradigma de *aprendizagem-ativa* de formar a atacar os problemas elencados anteriormente e de forma a propor uma disciplina factível a realidade da Universidade de Brasília (UnB).

1.1 Objetivo

O objetivo deste trabalho é propor para a disciplina de *Algoritmos e Programação de Computadores* um modelo de curso prático de programação.

Busca-se, por meio desta proposta de curso, oferecer um modelo pedagógico mais eficiente para o ensino da programação na linguagem C e o básico de circuitos embarcados. Tal proposta tem como motivação buscar tratar do problema da usual baixa profundidade na aprendizagem de programação e o problema da desmotivação nos alunos, causada por diversos fatores, dentre eles a separação artificial entre as disciplinas.

Neste trabalho são expostos XXXXXXXXXXXX planos de aula prática, que contêm uma explicação aprofundada de todos conceitos tratados no laboratório - tanto os conceitos diretamente ligados a programação na linguagem C, quanto os conceitos ligados a circuitos eletrônicos. Em todas descrições do plano de laboratório são também sugeridas formas de organização pedagógicas para ampliar e aprofundar a aprendizagem dos alunos.

1.2 Apresentação do manuscrito

No capítulo 2 a placa Intel[®] Galileo é apresentada e todos conceitos relacionados a seus componentes são explicados. São apresentados também as teorias educacionais que dão suporte ao método de *educação ativa*. Por fim, ainda no capítulo 2, é feita a proposta de reformulação da disciplina *Algoritmos e Programação de Computadores* apresentando um possível plano de ensino para tal disciplina.

Em seguida, o capítulo 3 apresenta as práticas laboratórias planejadas. Nessas práticas, é exposta a lista de conceitos de programação e eletrônica básica tratados, os materiais necessários, os esquemáticos dos circuitos e os códigos a serem utilizados com a placa Intel[®] Galileo.

Capítulo 2

Revisão Bibliográfica

O objetivo deste capítulo é explicitar todos conceitos relevantes deste trabalho relativos a ensino e aprendizagem de Ciências Exatas e Engenharia e a estrutura detalhada da placa Intel Galileu. Esses estudos servirão de base para a proposição um modelo de aula de programação básica para estudantes de graduação levando em conta os paradigmas de educação mais eficientes dentre os elencados.

2.1 Placa Intel[®] Galileo

A placa Intel[®] Galileo é uma *placa de desenvolvimento* com microcontrolador baseado processador Intel[®] Quark SoC X1000[10]. A placa Galileo possui software e hardware compatível com a placa *Arduino* com relação aos pinos digitais e analógicos. Um programa escrito para Arduino pode ser usado no Galileu por causa dessa compatibilidade. As Figura 2.1 e 2.2 mostram a placa Intel[®] em suas visão frontal e traseira.

Nesta seção são apresentadas, enumeradas e explicadas todas características da placa Intel[®] Galileo.

Primeiramente são apresentados os pinos da placa Galileo juntamente com uma breve descrição de seu uso. Após isso são descritas as enumeradas e explicadas todas características eletroeletrônicas da placa. Para cada tecnologia na placa é reservada uma pequena sub-seção neste capítulo para sua devida elucidação.

2.1.1 Pinagem da placa Intel[®] Galileo

Os pinos da placa Galileo nas partes frontal e traseira são mostrados nas figuras 2.1 e 2.2

A descrição de cada um desses pinos é a descrita na tabela 2.1:

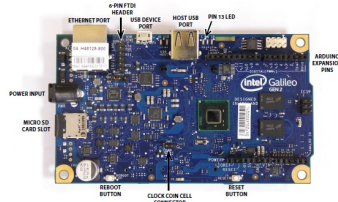


Figura 2.1: Descrição dos pinos da placa Galileo - parte frontal[1]

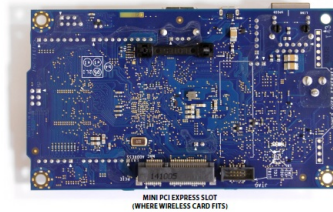


Figura 2.2: Descrição dos pinos da placa Galileo - parte traseira[1]

2.1.2 Características Elétricas e Eletrônicas da placa Intel[®] Galileo

As características elétricas e eletrônicas da placa são enumeradas a seguir. As características que têm uma sub-seção para explicação mais aprofundada estão marcadas com *itálico* e **negrito**:

- Clock de 400 MHz
- Arquitetura Intel[®] 32 bits
- 14 pinos digitais para entrada e saída, 6 das quais podem ser usadas para saída ***PWM***
- 6 pinos para entrada analógica utilizando o ***conversor analógico-digital AD7298***
- Barramento Serial ***I2C***
- Comunicação serial com periféricos ***SPI***
- Porta Serial ***UART***
- 16KBytes de memória ***L1 Cache***
- 512KBytes de memória ***SRAM***
- Clock de tempo real integrado (***RTC***)
- Barramento ***PCI Express***
- Conexão para ***USB Host e USB Client***
- 10 pinos padrões ***JTAG*** para debug
- 256 MBytes de memória ***DRAM***
- 11 KBytes de memória ***EEPROM***

Tabela 2.1: Pinos Galileo[1]

Pino	Descrição
Micro SD Card Slot	Pino no qual se pode um SD Card para permitir ao Galileo a execução de uma versão de Linux com mais recursos
Arduino Expansions Pins	Pinos de entrada e saída da placa Galileo. Esses pinos são compatíveis com os pinos do Arduino e Shields relacionadas.
USB Device Port	Pino para conectar um cabo USB do Galileo ao computador para carregar o Galileo com um programa Arduino
Host USB Port	Pino para conectar um dispositivo periférico (como webcam, caixa de som, etc)
6-Pin FTDI Header	Adaptador para comunicação serial computador - Linux instalado no Galileo
Power Input	Conexão para bateria de 12V. ATENÇÃO, a bateria sempre deve ser conectada ao Galileo antes de conectar um cabo USB do Galileo ao computador para evitar danos a placa.
Ethernet Port	Pino para conectar o Galileo à Internet pelo cabo Ethernet
Mini PCI Express Slot	Pino para conectar um cartão WiFi
Clock Battery Power	Conexão para uma bateria de relógio de 3V de forma a fazer com Galileo guarde informações de data e hora
Reboot Button	Botão para realizar a placa, inclusive o sistema operacional
Reset Button	Botão para resetar o código que foi carregado no Galileo

2.1.2.1 Sinal PWM

Pulse Width Modulation (PWM) ou Modulação por Largura de Pulso é uma técnica que modulação de impulso utilizada principalmente para codificar uma mensagem num sinal pulsante[11]

Para o caso do Intel Galileo, as aplicações do sinal PWM são principalmente relacionadas ao controle da tensão DC fornecida a um circuito.

O sinal PWM é gerado com ondas quadradas, de período T de ciclo. Durante parte do período, o sinal terá amplitude V_{max} . O intervalo de tempo no qual no sinal tem amplitude V_{max} é chamado *Duty-Cicle* como mostra a Figura 2.3. O valor DC de um sinal periódico é calculado como a média aritmética da amplitude do sinal no período. O valor da tensão DC fornecida ao circuito pelo Sinal PWM é calculado com a equação 2.1:

$$V_{dc} = 1/T() \int_0^{DutyCicle} V_{max} dt + \int_{DutyCicle}^T V_{min} dt) \quad (2.1)$$

$$V_{dc} = 1/T(DutyCicle * V_{max} + T * V_{min} - DutyCicle * V_{min}) \quad (2.2)$$

Se a tensão mínima (V_{min}) for igual a zero, o valor DC do sinal PWM é dado por:

$$V_{dc} = \frac{DutyCicle * V_{max}}{T} \quad (2.3)$$

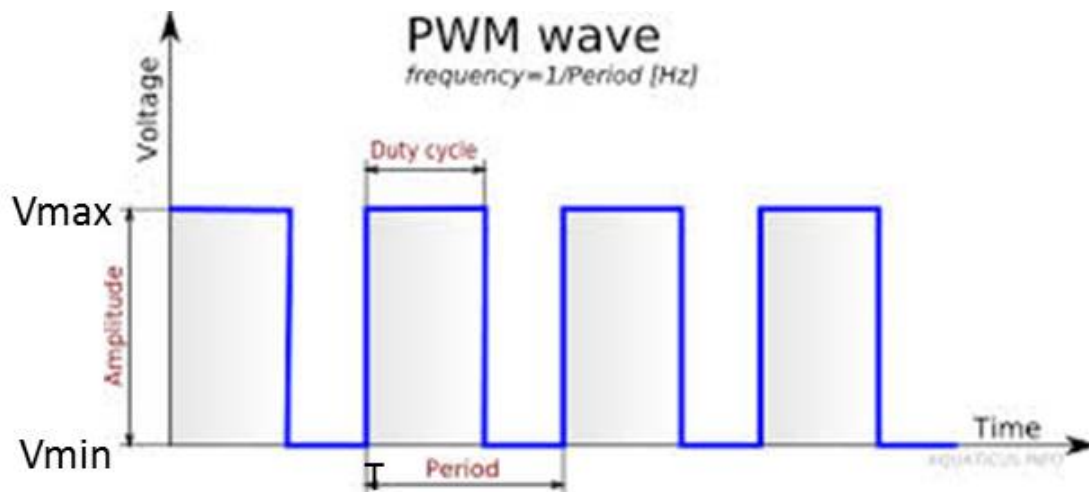


Figura 2.3: Sinal PWM

A equação 2.3 mostra que quanto maior for o tempo que o sinal permanecer no seu valor máximo (V_{max}), mais próximo de V_{max} será o valor DC fornecido ao circuito.

O sinal PWM é gerado na placa Galileo utilizando o clock interno máximo de 400 MHz e registradores de Timer específicos para contagem de pulsos do clock.

Como exemplo para a geração do sinal PWM, digamos que o clock da placa foi setado para a frequência 1kHz. Isso significa que a cada 1ms, o clock gerará um pulso, como indicado na equação 2.4.

$$f = 1Khz \rightarrow T = 1ms \rightarrow 1000 \text{ pulsos de clock por segundo} \quad (2.4)$$

Como a tensão de operação da placa Galileo é 5 V, então:

$$V_{max} = 5V \quad (2.5)$$

Caso se queria gerar um sinal PWM cujo componente DC seja 2.5, é necessário então que durante metade do ciclo do sinal PWM, a amplitude do sinal seja 5 V e durante a outra metade do ciclo, a amplitude seja 0V. Para criar tal sinal, o microcontrolador realiza contagem de pulsos de clock.

Para gerar 2.5 V, o microcontrolador (para a frequência exemplo de 1kHz) realiza a contagem de 500 pulsos de clock no intervalo de *DutyCycle* e realiza, após isso, a contagem de 500 pulsos no período no qual a amplitude será de 0 V. Dessa forma, é gerado digitalmente o sinal PWM na placa Galileo.

Como dito no início desta seção, placa Galileo é compatível com a placa Arduino, tanto a nível de hardware quanto a nível de software. Daí, para executar a criação de um sinal PWM na placa

galileu deve-se chamar a função *analogWrite(int porta, int valor)*.

A função *analogWrite(int porta, int valor)* recebe como parâmetros dois inteiros. O inteiro *porta* indica quais dos pinos digitais, habilitados para saída PWM, foi selecionado. O inteiro *valor* deve ser um inteiro entre 0 e 255.

```
1 //Comando para setar na porta digital 5 o valor 5*(127/255) = 2.5 Volts
2 analogWrite(5, 127);
```

A tensão DC que estará presente no pino digital segue a formula: 2.6

$$V_{dc} = \frac{5 * valor}{255} \quad (2.6)$$

2.1.2.2 Conversão analógico-digital

2.1.2.3 Barramento Serial I2C

2.1.2.4 Comunicação serial SPI

2.1.2.5 Porta Serial UART

2.1.2.6 Memória Cache

2.1.2.7 Memória SRAM

2.1.2.8 Clock de tempo real - RTC

2.1.2.9 Barramento PCI-Express

2.1.2.10 USB Host e USB Client

2.1.2.11 Pinos JTAG

2.1.2.12 Memória DRAM

2.1.2.13 Memória EEPROM

Capítulo 3

Desenvolvimento

Resumo opcional.

3.1 Introdução

Na introdução deverá ser feita uma descrição geral da metodologia que foi seguida para o desenvolvimento. A seguir, é feita a descrição do sistema desenvolvido.

Deve-se ressaltar que equações fazem parte do texto, devendo receber pontuação apropriada e ser numerada. Alguns exemplos são mostrados na seção 3.3.

3.2 Arquitetura geral

3.3 Classificador estatístico de padrões

O classificador automático de padrões utiliza o princípio da menor distância no processo de associação de dados. Assim, sendo \mathbf{x} o vetor de características extraídas de uma imagem e $\mathbf{P}_{\mathbf{x}}$ sua matriz de covariâncias respectiva, utiliza-se

$$d_i = (\mathbf{x} - \mathbf{p}_i)^T (\mathbf{P}_{\mathbf{x}} + \mathbf{P}_{\mathbf{p}_i})^{-1} (\mathbf{x} - \mathbf{p}_i) \quad (3.1)$$

como métrica para a distância estatística de \mathbf{x} e um padrão de características \mathbf{p}_i e matriz de covariâncias $\mathbf{P}_{\mathbf{p}_i}$. Esta métrica é conhecida também pela denominação “distância de Mahalanobis”. A distância definida pela Eq. (3.1) segue distribuição χ_n^2 , em que n é a dimensão da base do vetor \mathbf{x} . Assim sendo, no caso específico de $n = 3$, o padrão associado ao vetor \mathbf{x} é dito casado com o padrão \mathbf{p}_i com 5% de margem de erro se

$$d_i \leq 7,815. \quad (3.2)$$

A seguir é mostrado como o modelo Latex apresenta os comandos `\section` `\subsection` e `\subsubsection`. Por questão de estilo, o texto deve ser organizado de modo a se evitar o uso de `\subsubsection`.

3.4 Seção

Meu texto da seção.

3.4.1 Sub-seção

Meu texto da sub-seção.

3.4.1.1 Sub-sub-seção

Meu texto da sub-sub-seção.

Se necessário, use notas de rodapé ¹

¹Essa é uma nota de rodapé.

Capítulo 4

Resultados Experimentais

Resumo opcional.

4.1 Introdução

Na introdução deverá ser feita uma descrição geral dos experimentos realizados.

Para cada experimentação apresentada, descrever as condições de experimentação (e.g., instrumentos, ligações específicas, configurações dos programas), os resultados obtidos na forma de tabelas, curvas ou gráficos. Por fim, tão importante quando ter os resultados é a análise que se faz deles. Quando os resultados obtidos não forem como esperados, procurar justificar e/ou propor alteração na teoria de forma a justificá-los.

4.2 Avaliação do algoritmo de resolução da equação algébrica de Riccati

O algoritmo proposto para solução da equação algébrica de Riccati foi avaliado em diferentes máquinas. Os tempos de execução são mostrados na Tabela 4.1. Nesta tabela, os algoritmos propostos receberam a denominação CH para Chandrasekhar e $CH + LYAP$ para Chandrasekhar com Lyapunov. As implementações foram feitas em linguagem *script* MATLAB.

Observa-se que o algoritmo $CH + LYAP$ apresenta tempos de execução superiores com relação ao algoritmo CH . Entretanto, era esperado que o algoritmo CH fosse mais rápido. Este resultado

Tabela 4.1: Tempos de execução em segundos para diferentes máquinas

Algoritmo	Laptop 1.8 GHz	Desktop PIII 850 MHz	Desktop MMX 233	Laptop 600 MHz
Matlab ARE	649,96	1.857,5	7.450,5	9.063,9
CH	259,44	606,4	2.436,5	2.588,5
$CH + LYAP$	357,86	952,9	3.689,2	3.875,0

se justifica pelo fato de o algoritmo CH fazer uso de funções embutidas do MATLAB. Já o algoritmo $CH + LYAP$ faz uso também de funções *script* externas, aumentando bastante seu tempo computacional.

Capítulo 5

Conclusões

Este capítulo é em geral formado por: um breve resumo do que foi apresentado, conclusões mais pertinentes e propostas de trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] S. BURFOOT J., G. D. e. H. C. B. *A Teacher's Guide to Intel Galileo*. Buildind C5B, Macquarie University, North Ryde, NSW, 2109, 2015.
- [2] SOVIC, A.; JAGUST, T.; SERSIC, D. How to teach basic university-level programming concepts to first graders? In: *Integrated STEM Education Conference (ISEC), 2014 IEEE*. [S.l.: s.n.], 2014. p. 1–6.
- [3] COTO, M.; MORA, S.; ALFARO, G. Giving more autonomy to computer engineering students: Are we ready? In: *IEEE Global Engineering Education Conference, EDUCON 2013, Berlin, Germany, March 13-15, 2013*. [s.n.], 2013. p. 618–626. Disponível em: <<http://dx.doi.org/10.1109/EduCon.2013.6530170>>.
- [4] CELETI, F. R. Origem da educação obrigatória: Um olhar sobre a prússia. *Revista Saber Acadêmico*, v. 1, n. 1, p. 29–33, June 2012.
- [5] OLIVER, J.; TOLEDO, R. On the use of robots in a pbl in the first year of computer science / computer engineering studies. In: *Global Engineering Education Conference (EDUCON), 2012 IEEE*. [S.l.: s.n.], 2012. p. 1–6. ISSN 2165-9559.
- [6] F. de O. V. Crescimento, evolução e o futuro dos cursos de engenharia. *Revista de Ensino de Engenharia*, v. 24, n. 2, p. 3–12, December 2005.
- [7] R., W. *Alta taxa de desistência na universidade causa déficit de engenheiros*. Setembro 2013. [Online; posted 4-Setembro-2013].
- [8] LAHTINEN, E.; ALA-MUTKA, K.; JÄRVINEN, H.-M. A study of the difficulties of novice programmers. *SIGCSE Bull.*, ACM, New York, NY, USA, v. 37, n. 3, p. 14–18, jun. 2005. ISSN 0097-8418. Disponível em: <<http://doi.acm.org/10.1145/1151954.1067453>>.
- [9] ESCUDERO, M. R.; HIERRO, C. M.; PABLO, A. Pérez de Madrid y. Using arduino to enhance computer programming courses in science and engineering. In: *EDULEARN13 Proceedings*. [S.l.]: IATED, 2013. (5th International Conference on Education and New Learning Technologies), p. 5127–5133. ISBN 978-84-616-3822-2. ISSN 2340-1117.
- [10] INTEL. *DataSheet Intel Galileo Gen 2 Development Board*. [S.l.], 2014.
- [11] SEDRA., A. S.; SMITH, K. C. *Microeletronics Cicuits*. [S.l.]: Oxford University Press, 2004.

ANEXOS

I. DIAGRAMAS ESQUEMÁTICOS

II. DESCRIÇÃO DO CONTEÚDO DO CD