



UNIVERSIDADE DO VALE DO TAQUARI - UNIVATES  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

**PROPOSTA DE OTIMIZAÇÃO DE COMUNICAÇÃO PARA  
PLATAFORMAS MICROCONTROLADAS UTILIZANDO  
MAPEAMENTO EM MEMÓRIA EXTERNA**

Karolina Amanda Schubert

Lajeado, novembro de 2020



Karolina Amanda Schubert

**PROPOSTA DE OTIMIZAÇÃO DE COMUNICAÇÃO PARA  
PLATAFORMAS MICROCONTROLADAS UTILIZANDO  
MAPEAMENTO EM MEMÓRIA EXTERNA**

Trabalho de conclusão de curso apresentado no Centro de Ciências Exatas e Tecnológicas, da Universidade do Vale do Taquari – Univates, como parte dos requisitos para obtenção do título de bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Alexandre Stürmer Wolf

Lajeado, novembro de 2020.

## **AGRADECIMENTOS**

*Myself*, por ter abraçado minhas dificuldades e juntamente com a indústria farmacêutica ter encontrado forças para continuar.

Ao Israel André Fröhlich, por estar sempre ao meu lado, me incentivando e motivando.

Aos meus amigos Luan Araujo dos Santos e Tatiane Simoneti Vaz, pela amizade e por terem me ajudado nesta caminhada.

A Pretinha, minha cachorra, que sempre esteve comigo nos momentos de escrita.

Por último, e não menos importante, meu orientador Prof. Dr. Alexandre Stürmer Wolf, por ter acreditado em mim e no meu trabalho, por ter me guiado nesta etapa, me orientando no desenvolvimento e conclusão deste trabalho.

## RESUMO

Este trabalho apresenta um estudo para a otimização no desempenho de comunicação entre dispositivos físicos através de recursos de *hardware* dedicado. Foram analisadas soluções atuais em termos de *hardware* e *software* para a plataforma de prototipação eletrônica Arduino. Para a aplicação do estudo, foi desenvolvida uma placa para testes e posteriormente comparada com soluções existentes no mercado. A placa de prototipação utilizada no estudo utiliza um microcontrolador ATmega2560, que possui pinos de entradas e saídas para sinais externos, sendo que alguns destes apresentam recursos de *hardware* dedicado. Foram analisadas as formas de comunicação do microcontrolador com *hardware* externo utilizando estruturas diferentes de endereçamento através de pinos de uso geral e pinos com *hardware* gerenciado por registradores, possibilitando o mapeamento em memória externa - XMEM. Para demonstrar a otimização no desempenho na comunicação utilizando os pinos com *hardware* gerenciado, foi desenvolvido uma *shield* para acoplar ao Arduino Mega 2560 que explora os pinos de *external memory interface* do microcontrolador. As análises basearam-se nas variações de tempo de inicialização das placas *shields*, comercial e desenvolvida, com *display* LCD. Ao final deste projeto, conclui-se que a partir das análises realizadas, a otimização do *hardware* e *software* obteve uma redução de 18 ms para 6,94 ms, alcançando um aprimoramento de 61,45 %.

**Palavras-Chave:** Registradores de Memória; *Display* LCD; Arduino Mega 2560; Microcontrolador.

## **ABSTRACT**

This work presents a study for the optimization of the communication performance between physical devices through dedicated hardware resources. Current solutions in terms of hardware and software for the Arduino electronic prototyping platform were analyzed. For the application of the study, a test board was developed and later compared with existing solutions on the market. The prototyping board used in the study uses an ATmega2560 microcontroller, which has input and output pins for external signals, some of which have dedicated hardware resources. Ways of communication of the microcontroller with external hardware were analyzed using different addressing structures through pins of general use and pins with hardware managed by registers, enabling the mapping in external memory - XMEM. In order to demonstrate the optimization in communication performance using pins with managed hardware, a shield was developed to connect to the Arduino Mega 2560 that exploits the microcontroller external memory interface pins. The analyzes were based on variations in the startup time of shields, commercial and developed, with LCD display. At the end of this project, it is concluded that from the analysis carried out, the optimization of hardware and software resulted in a reduction from 18 ms to 6.94 ms, reaching an improvement of 61.45%.

**Key words:** Memory Recorders; LCD display; Arduino Mega 2560; Microcontroller.

## LISTA DE FIGURAS

Figura 01: Arquitetura de um microcontrolador.	16
Figura 02: ATmega2560.	17
Figura 03: Configuração da memória externa.	20
Figura 04: Arduino Mega 2560.	21
Figura 05: <i>Pinout</i> Arduino Mega 2560.	23
Figura 06: IDE Arduino.	24
Figura 07: <i>Shield</i> LCD acoplada no Arduino Mega 2560.	25
Figura 08: Estrutura do <i>hardware</i> .	27
Figura 09: Conjunto de fendas do VECC.	27
Figura 10: Exemplos de periféricos da plataforma MISL-ASE.	28
Figura 11: Conexão do microcontrolador e as memórias EPROM e RAM.	29
Figura 12: Sistema sendo mostrado em dois estados.	30
Figura 13: Esquemático da <i>shield</i> LCD.	33
Figura 14: Board da <i>shield</i> LCD.	33
Figura 15: <i>Shield</i> LCD.	34
Figura 16: LCD Keypad Shield.	35

Figura 17: Medição da análise “A”.	41
Figura 18: Medição da análise “B”.	42
Figura 19: Medição da análise “C.1”.	43
Figura 20: Medição da análise “C.2”.	44
Figura 21: Medição da análise “D.1”.	45
Figura 22: Medição da análise “D.2”.	46
Figura 23: Medição da análise “E”.	47

## LISTA DE QUADROS

Quadro 01: Comparativos dos estudos relacionados.	32
Quadro 02: Pinagem <i>display</i> LCD.	38
Quadro 03: Ligação entre o Arduino Mega 2560 e a LCD Keypad shield.	39
Quadro 04: Ligação entre o Arduino Mega 2560 e a <i>shield</i> desenvolvida.	40
Quadro 05: Comparação dos testes realizados.	49

## LISTA DE ABREVIATURAS, SÍMBOLO E SIGLAS

A/D	Analógico e Digital
CI	Circuito Integrado
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
EEPROM	Electrically-Erasable Programmable Read-Only Memory
E/S	Entrada e Saída
GPIO	General Purpose Input/Output
LCD	Liquid Crystal Display
MIPS	Milhões de Instruções Por Segundo
RISC	Reduced Instruction Set Computer
RAM	Random Access Memory
SRAM	Static Random-Access Memory
XMEM	eXternal MEMory
XMCRA	External Memory Control Register A
XMCRB	External Memory Control Register B

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
1.1 Tema	12
1.1.1 Delimitação do tema	13
1.2 Objetivo geral	13
1.2.1 Objetivos específicos	13
1.3 Justificativa	14
1.4 Estrutura	14
<b>2 REVISÃO TEÓRICA</b>	<b>15</b>
2.1 Microcontrolador	15
2.2 ATmega2560	16
2.3 Mapeamento dos dispositivos em memória	18
2.4 Arduino Mega 2560	21
2.4.1 Características de Hardware	21
2.4.2 Características de Software	23
2.5 Shields - Placas de expansão	24
<b>3 TRABALHOS RELACIONADOS</b>	<b>26</b>
3.1 Estudo 1	26
3.2 Estudo 2	27
3.3 Estudo 3	28
3.4 Estudo 4	29
3.5 Estudo 5	30
3.6 Comparativo entre os Trabalhos Relacionados	31
<b>4 MATERIAIS E MÉTODOS</b>	<b>32</b>
4.1 Pesquisa científica	33
4.1.1 Abordagem	33
4.1.2 Natureza	33
4.1.3 Objetivos	33
4.1.4 Procedimentos	33

4.2 Tecnologias	33
4.2.1 Desenvolvimento da shield LCD	34
4.2.2 LCD Keypad Shield	36
4.2.3 Display LCD	37
4.2.4 Software	38
4.2.5 Verificação do tempo de inicialização do hardware	39
<b>5 RESULTADOS</b>	<b>42</b>
5.1 Teste utilizando a LCD Keypad Shield	42
5.2 Teste utilizando a shield desenvolvida	44
5.3 Comparação dos resultados	47
<b>6 CONCLUSÃO</b>	<b>49</b>
<b>REFERÊNCIAS</b>	<b>50</b>
<b>APÊNDICES E ANEXOS</b>	<b>53</b>

## 1 INTRODUÇÃO

Diversas aplicações de sistemas digitais, como eletroeletrônicos e eletrodomésticos, apresentam internamente microcontroladores, sendo eles os responsáveis por executar tarefas específicas de funcionalidade destes equipamentos (AURELIANO,2017). Devido a esta versatilidade existem diversos fabricantes, cada um deles comercializa modelos com especificações e características distintas, com isso a escolha do microcontrolador depende da aplicação em que será utilizado, portanto é fundamental conhecer as características deste componente (GREFIM, SILVA E CRUZ, 2018).

Com o intuito de otimizar aplicações que dispõem de um microcontrolador é necessário encontrar uma sintonia entre a forma em que o código é desenvolvido e o *hardware* no qual é realizado esta operação. Em diversas aplicações, tal sintonia requer um diagnóstico mais detalhado da performance da implementação original. Para executar o diagnóstico é necessário conhecer os componentes que constituem o *hardware* e a forma que eles aplicam suas instruções (CORRÊA, 2020).

Aureliano (2017), compara um microcontrolador como uma espécie de computador, pois apresenta uma unidade central de processamento (CPU, do inglês *Central Processing Unit*), memória para armazenar o programa que será executado e memória de armazenamento de variáveis. Alguns modelos apresentam periféricos para comunicação, conversões, entre outras funcionalidades que dependem da marca ou modelo.

Em relação à memória dos microcontroladores, Patterson e Hennessy (2011), atribuem uma arquitetura hierárquica. Esta hierarquia é pertinente ao envio de dados entre os diferentes níveis de memória, de modo mais simples, quais os trajetos existentes para que os dados sejam transportados de um ponto ao outro dentro desta hierarquia. Esta ordenação motiva-se pela existência de diferentes tipos de memória, que apresentam tamanhos, velocidades e acessos distintos. Os autores complementam dizendo que a CPU realiza o acesso e as manipulações dos dados que estão na memória. Menor será o tempo deste acesso quanto mais adjacente for o nível da memória em que estes dados encontram-se.

Corrêa (2020), utiliza a expressão de “malha” para a hierarquia, ele afirma que o uso inadequado dela pode provocar um consumo excessivo do tempo de atuação da CPU, resultando numa ineficiência à aplicação. Ele ressalta a necessidade em otimizar estes processos, envio e/ou leitura de dados à memória, ponderando uma investigação detalhada no código fonte utilizado. Esta análise leva em consideração a estrutura do programa, especialmente a forma de utilização das chamadas dos dados e suas sub rotinas.

O autor complementa que, na obtenção deste diagnóstico, quanto maior o nível de detalhamento das informações de desempenho mais fácil torna-se a tarefa do programador em aprimorar o código. Determinadas situações requerem métodos e equipamentos adicionais, como o osciloscópio, para a realização de um diagnóstico mais efetivo, a fim de realizar a otimização.

## **1.1 Tema**

O tema do presente trabalho tem como objetivo realizar a análise de desempenho e proposta de melhorias em *hardware* e *software* para uma plataforma computacional microcontrolada através da utilização de registradores de memória externa.

### 1.1.1 Delimitação do tema

O estudo utiliza o microcontrolador ATmega2560, que está presente nas placas Arduino Mega 2560.

A análise foi realizada através da comparação entre o tempo em que uma placa expansora (*shield*) comercial leva para iniciar e executar os comandos utilizando pinos GPIO, comparado com uma placa expansora que foi desenvolvida e explora os pinos com recursos de registradores para o mapeamento em memória externa XMEM.

## 1.2 Objetivo geral

Este trabalho tem como objetivo geral a comparação das soluções comerciais atuais, em termos de *hardware* e *software* para a plataforma Arduino Mega 2560 além de sugestões para melhorar o desempenho e a utilização destas soluções.

### 1.2.1 Objetivos específicos

- Comparar as soluções comerciais de *displays* LCD para a plataforma Arduino Mega 2560;
- Analisar o desempenho das soluções de *software* que utilizam o *hardware* de placas comercializadas;
- Desenvolver um *shield* com *display* LCD, que utilize os recursos da XMEM do ATmega2560;
- Comparar o desempenho de soluções comerciais com a *shield* desenvolvida.

### **1.3 Justificativa**

Ao longo da trajetória acadêmica utilizam-se em várias disciplinas as placas comerciais da empresa Arduino, para a execução de trabalho e projetos acadêmicos. Com a consolidação destas placas e da plataforma de programação, não só na área acadêmica como na área comercial, é facilmente encontrado uma vasta disponibilidade de documentação e exemplos de aplicabilidade, favorecendo a utilização do Arduino no ambiente acadêmico.

### **1.4 Estrutura**

Este trabalho é composto por 6 capítulos em que se organizam da seguinte forma: o primeiro capítulo apresenta uma breve introdução sobre microcontroladores; o capítulo 2 apresenta a revisão teórica com conceitos estudados e que constituem o desenvolvimento do protótipo; o capítulo 3 descreve artigos relacionados com esta monografia, apresentado um comparativo com o que cada um deles contempla ou não contempla; o capítulo 4 apresenta a proposta, os materiais e métodos que serão aplicados para a análise de tempo; o capítulo 5 expõe os resultados obtidos através das análises envolvendo o *hardware* e *software* deste trabalho; por último, o capítulo 6 apresenta a conclusão da monografia, ressaltando algumas das considerações finais do projeto.

## **2 REVISÃO TEÓRICA**

Será apresentado neste capítulo o embasamento teórico necessário para a compreensão do trabalho. Serão abordados temas como microcontroladores, mapeamento de memória, microcontrolador ATmega2560.

### **2.1 Microcontrolador**

Em meados da década de 70, desenvolvido por um time da empresa Texas Instrument surgem os primeiros microcontroladores, derivados dos microprocessadores, inicialmente eles possuíam a função de realizar cálculos. Com aprimoramentos o componente tomou-se capaz de tomar decisões e posteriormente lhe foi incorporado memórias e outras arquiteturas (SANTOS e LARA JUNIOR, 2019).

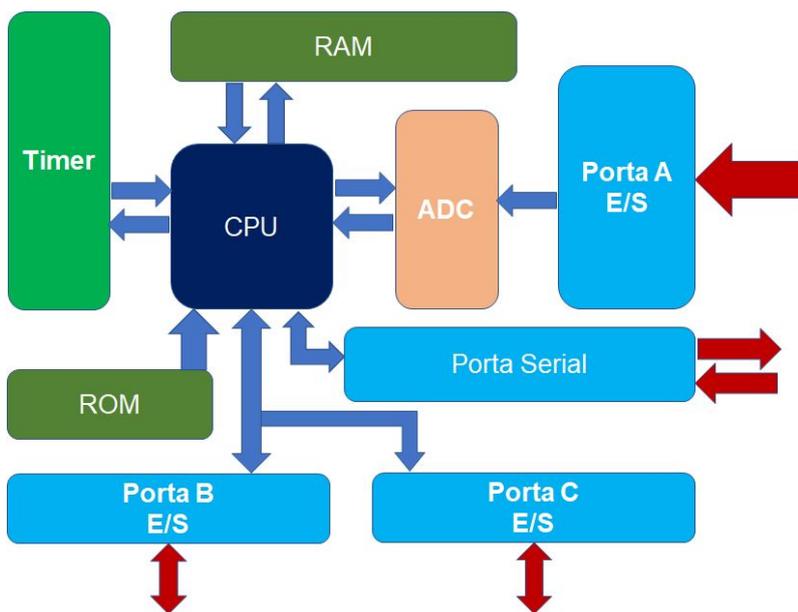
Medeiros (2018), define o microcontrolador como um circuito integrado (CI), capaz de realizar processamentos lógicos com alta precisão e rapidez. Este CI possui a vantagem de uma ampla viabilidade de aplicações, tornando-o maleável e possibilitando ajustes para atender às instruções que lhe são atribuídas. Para Grefim, Silva e Cruz (2018), o microcontrolador é circuito integrado simples que apresenta sistemas eletrônicos que possuem circuitos especialistas e funções empregadas em sua construção. Este CI é composto por um núcleo processador que é responsável pelo processamento de cálculos lógicos e aritméticos,

gerenciamento da memória e os comandos de seus periféricos de entradas e saídas programáveis.

Grefim, Silva e Cruz (2018), também apontam o microcontrolador como um “Dispositivo versátil” de fácil programação, baixo custo e tamanho condensado, amplamente utilizado em diversas aplicações industriais, domésticas entre outras.

A partir dessas definições, conclui-se que os microcontroladores foram planejados com o intuito de reduzir a quantidade de componentes externos nas aplicações, visto que possui unidade de processamento, memória, dispositivos de comunicação serial, entre outros. Obtendo então um valor reduzido tanto no custo como de tempo e espaço na elaboração e criação de dispositivos. A figura 01 ilustra a arquitetura interna de um microcontrolador.

Figura 01: Arquitetura de um microcontrolador.



Fonte: do autor, 2020.

## 2.2 ATmega2560

Fabricado pela ATMEL, atualmente pertencente a empresa Microchip Technology, o microcontrolador ATmega2560 apresentado na figura 02, é um

circuito integrado CMOS<sup>1</sup> de 8 *bits* e de baixa potência, que utiliza a arquitetura AVR, sendo própria da ATMEL, baseada em uma arquitetura RISC<sup>2</sup>, esta estrutura incrementa um conjunto simples e pequeno de comandos que para serem executados necessitam aproximadamente a mesma quantidade de tempo. O CI executa os comandos em um único ciclo de *clock*, alcançando *throughputs* (quantidade de dados transferidos de um lugar a outro) de aproximadamente 1 MIPS (Milhões de Instruções Por Segundo) a 1 MHz, o que possibilita o programador otimizar o consumo de energia em relação a velocidade do processamento (ATMEL, 2014).

O ATmega2560 possui 256 kB de memória *Flash*, 4 kB de memória EEPROM (*Electrically-Erasable Programmable Read-Only Memory*), não volátil, e 8 kB de memória RAM (*Random Access Memory*), volátil, operando na frequência de 16 MHz. O microcontrolador apresenta a dimensão físicas de 16,25 mm de largura, 16,25 mm de comprimento e 1,2 mm de altura (ATMEL, 2014).

Figura 02: ATmega2560.



Fonte: Autor, 2020.

---

<sup>1</sup> CMOS: A sigla vem do inglês, *Complementary metal-oxide semiconductor*, traduzindo, metal-óxido-semicondutor complementar. Isso define a característica física do circuito integrado, tais componentes possuem um baixo consumo de energia (WIDMER, 2018).

<sup>2</sup> RISC: É uma arquitetura que possibilita um conjunto simples e pequeno de comandos que têm cerca de a mesma quantidade de tempo para serem executadas. Acrônimo para *Reduced Instruction Set Computer* (TANENBAUM, 2013).

A anexo 01 apresenta o *pinout* do microcontrolador, e no anexo 02 consta a tabela com os pinos do ATmega2560 e suas respectivas funções.

### 2.3 Mapeamento dos dispositivos em memória

A memória é o local em que os códigos de comando e os dados ficam armazenados, ela é formada por *bytes*, cada um deles pode armazenar uma informação. Os *bytes* são referenciados por um valor, este é chamado de endereço de memória (TANENBAUM, 2013). O autor cita, em 2009, que a memória é acessada pela CPU a qual busca informações e as executa. A CPU possui um ciclo de execuções, ela acessa a primeira instrução da memória a decodifica e executa, seguindo então para a instrução seguinte.

O autor afirma que todas as CPUs possuem registradores internos, eles são responsáveis pelo armazenamento de variáveis essenciais, como também de resultados temporários. Com isso, comumente os conjuntos de comandos possuem instruções para carregar uma variável da memória para um registrador e armazenar uma variável de um registrador na memória. Quando há a necessidade de alteração do valor presente na memória é necessário carregá-lo em um registro, e então modificá-lo para que possa ser atualizado na memória. “O uso do registrador para armazenar uma variável oferece duas vantagens: o acesso é mais rápido e são necessários menos *bits* para se localizar o operando” (TANENBAUM, 2013).

Através da escrita em registradores, a CPU pode controlar dispositivos de entrada e saída (E/S), enviar ou receber dados ou executar outras tarefas. A CPU também é capaz de realizar tanto a leitura de dados como também o armazenamento em memória de registradores especiais, como por exemplo, os registradores *PORTx*<sup>3</sup> (TANENBAUM, 2013).

Para Rajewski (2018), o mapeamento em memória remete ao seu endereçamento dos dispositivos de entrada e saída e as instruções da CPU. Para

---

<sup>3</sup> Registrador PORT: É o registrador responsável pela escrita e leitura do nível dos pinos associados a uma porta (ATMEL, 2014).

acomodar os dispositivos de E/S, áreas do espaço endereçável da CPU devem ser reservadas. A reserva de espaço pode ser temporária ou permanente. Cada dispositivo de E/S monitora a via de endereçamento da CPU e responde a qualquer acesso dela ao espaço de endereço destinado ao dispositivo, conectado a via de dados ao registrador do *hardware* do dispositivo desejado.

O autor ainda cita que o mapeamento da memória é comumente realizado no processador, exemplificando com os registradores *PORTx* de um controlador ATmega, em que há a possibilidade de escrita do código:

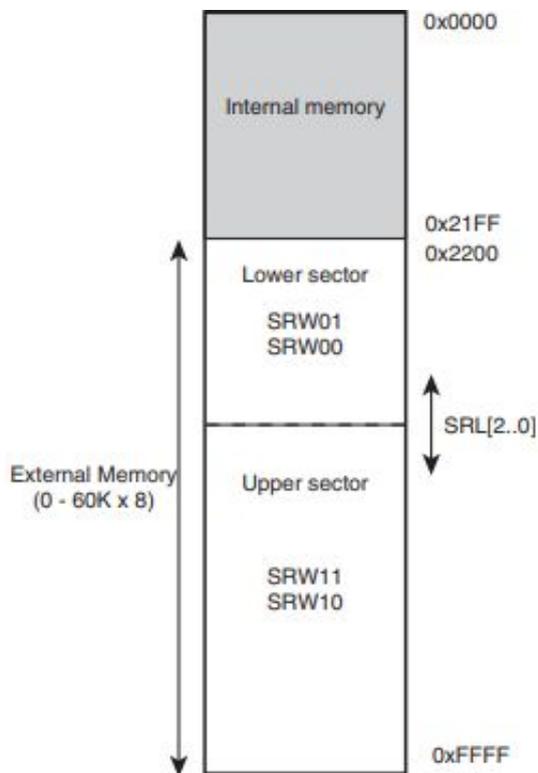
```
PORTB = 0XAA;
```

A linha do código define os oito pinos de E/S designados para PORTB para o valor 0xAA. PORTB é na verdade apenas uma macro e é realmente um ponteiro para um endereço de memória especial. Esse endereço na memória não apenas mapeia para a RAM, mas também mapeia para um periférico de entrada e saída que obtém o valor e o envia para os pinos de E/S. Sem esse mapeamento na memória, o microcontrolador seria incapaz de inserção e envio de dados (RAJEWSKI, 2018).

De acordo com a ATMEL (2014), alguns dos dispositivos disponibilizam a *interface* de memória externa (*External Memory Interface*), como por exemplo o ATmega2560 (seção 2.2). Esta interface permite realizar a comunicação com outros dispositivos de memória, assim como outros periféricos (*Display* LCD, conversores A/D). Estes dispositivos são conectados através de GPIOs (*General Purpose Input/Output*), pinos de entrada e saída que são programáveis em que os dados utilizados viabilizam esta *interface* entre o microcontrolador e o periférico.

A figura 03 apresenta a configuração da memória externa XMEM (*eXternal MEMORY*), quando ela está ativa o espaçamento de endereços que está fora da SRAM (memória estática de acesso randômico - *Static Random Access Memory*) ficam disponíveis, utilizando os pinos dedicados da memória externa (ATMEL, 2014).

Figura 03: Configuração da memória externa.



Fonte: ATMEL, 2014

De acordo com a fabricante do microcontrolador ATmega2560, os *bits* de controle da *Interface* de Memória Externa localizam-se em dois registros, XMCR A (*External Memory Control Register A*) e o XMCR B (*External Memory Control Register B*). Quando a XMEM estiver ativa, sua *interface* irá sobrescrever a configuração na direção dos registradores PORTx que correspondem às portas dedicadas da memória externa.

Todas as portas do microcontrolador têm atribuições de leitura, modificação e escrita quando utilizadas como portas digitais de entrada e saída, entretanto quando a *interface* XMEM estiver habilitada, possuem as seguintes características (ATMEGA, 2014):

- Pinos 22 (AD0) à 29 (AD7): Barramento de endereços de ordem inferior multiplexado e barramento de dados;
- Pinos 30 (A15) à 37 (A8): Barramento de endereços de alta ordem (número configurável de *bits*);

- Pino 39 (ALE): Habilitador do *latch* de endereço;
- Pino 40 (RD): Pino para leitura;
- Pino 41 (WR): Pino para leitura e escrita.

## 2.4 Arduino Mega 2560

O Arduino Mega 2560 (figura 04), é uma placa que foi desenvolvida pela empresa Arduino, que possui *hardwares* e *software open source* (*software* de código aberto). A companhia surgiu no Instituto de Design de Interações de Ivrea no começo dos anos 2000 através de um projeto de pesquisa de Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis (ARDUINO, 2020).

Em 2005 com o intuito de auxiliar alunos que não possuíam experiência em eletrônica e/ou programação de microcontroladores foi desenvolvida a primeira placa Arduino, ela auxiliava a criação de protótipos funcionais. As placas Arduino são consideradas uma plataforma de computação embarcada, em que há interação com o ambiente por meio do *hardware* e *software* (ARDUINO, 2020).

Figura 04: Arduino Mega 2560.



Fonte: Arduino, 2020.

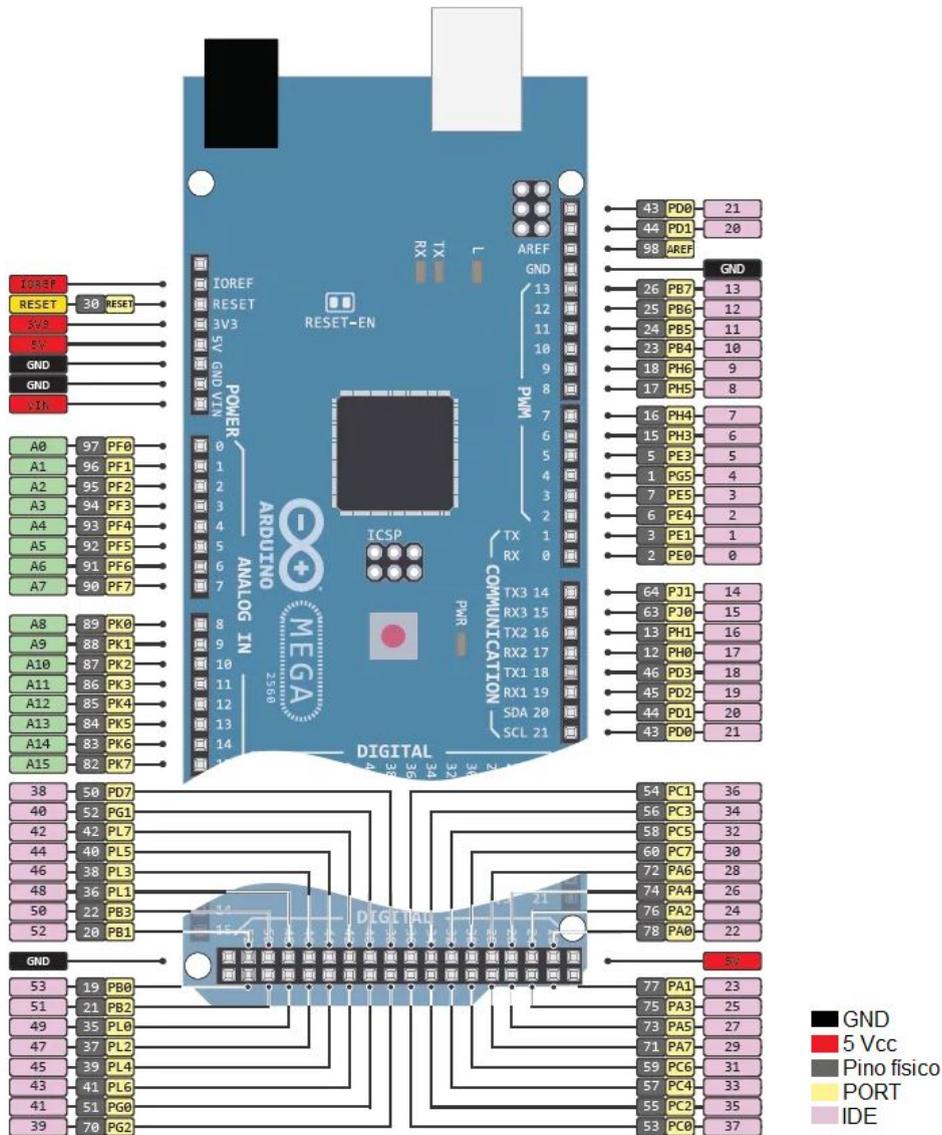
### 2.4.1 Características de *Hardware*

A placa Arduino Mega 2560 apresenta o microcontrolador ATmega2560. Para facilitar a aplicabilidade da placa embarcada, ela dispõe de 54 pinos digitais de

entrada e saída, em que 15 deles podem ser utilizados como saídas PWM (*Pulse Width Modulation* - modulação por largura de pulso), 16 pinos analógicos, 4 UARTs (*Universal Asynchronous Receiver/Transmitter*), um cristal oscilador de 16 MHz, conector USB, botão de *reset* (ARDUINO, 2020).

A figura 05 ilustra o *pinout* da placa, nota-se que os pinos físicos (identificados com a cor amarela) possuem uma nomenclatura diferente dos pinos do microcontrolador (identificados com a cor preta), entretanto ambos estão fisicamente conectados. Esta discrepância ocorre, pois, cada modelo de placa Arduino apresenta um microcontrolador diferente, mas o *software* de programação do Arduino, através de sua linguagem baseada em C++ possibilita uma programação otimizada dos pinos para os seus modelos de placas (ARDUINO, 2020).

Figura 05: Pinout Arduino Mega 2560.



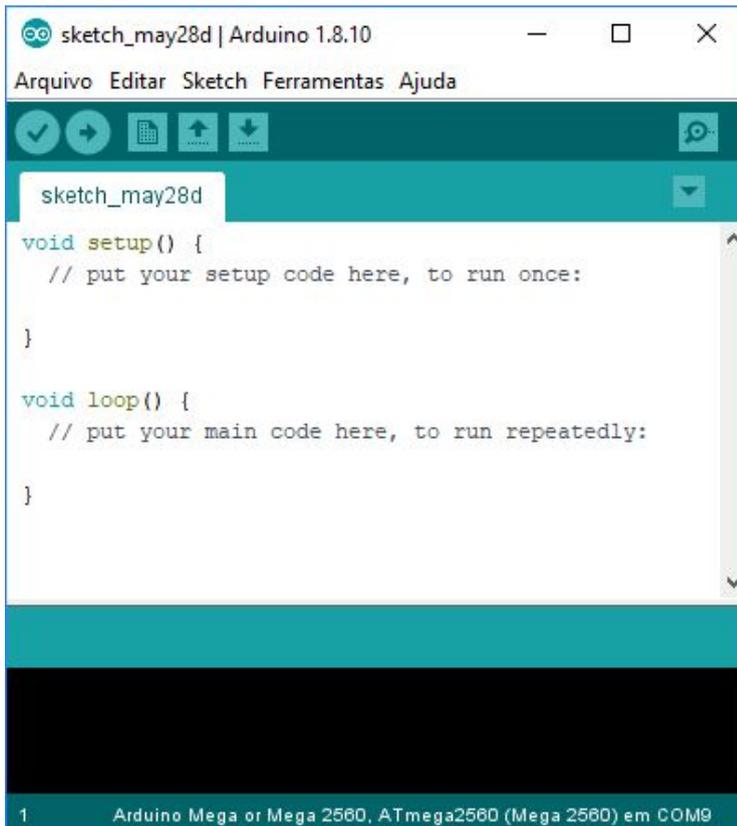
Fonte: Autor, 2020.

## 2.4.2 Características de Software

O software da empresa Arduino possui *interface* de programação que permite a compilação e o envio do código para qualquer umas das placas da empresa, ele pode ser executado em Windows, Mac OS X e Linux. Sua IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado), ilustrado na

figura 06, é de fácil programação e possibilita a escrita do código em C/C++ (ARDUINO, 2020).

Figura 06: IDE Arduino.



Fonte: Autor, 2020.

## 2.5 Shields - Placas de expansão

A placa Arduino Mega 2560 pode ser expandida através da utilização de uma *shield*. A *shield* é uma placa de circuito impresso que possui pinos de acoplamento. Podendo ser encaixada à placa do Arduino, expandindo suas funcionalidades e capacidades (SANTOS, 2018). O autor ainda ressalta que este tipo de expansão abre um leque de alternativas e inúmeras aplicações de uma forma simplificada, dispensando o uso de *jumpers* e *protoboard*.

A figura 07 mostra uma *shield* LCD com teclado acoplada em uma placa Arduino Mega 2560. De acordo com a fabricante DFROBOT o LCD *Keypad Shield*,

opera com a alimentação de 5 V, possui um *display* LCD, botões funcionam por meio de divisores de tensão comandados pela porta A/D do Arduino (cada nível de tensão é atribuído a um dos botões), potenciômetro para ajuste do *backlight*, pinos para *reset*, *enable*, e pinos para expansões.

Figura 07: *Shield* LCD acoplada no Arduino Mega 2560.



Fonte: Autor, 2020.

O *display* LCD presente nesta placa possui um controlador Hitachi HD44780, que possibilita ao programador um conjunto de instruções como: limpar a tela, posicionar o cursor, entre outras (CAMPOS, 2014). O anexo 03 ilustra o esquemático completo de ligação da *shield* LCD.

Nesta seção foi apresentado o referencial teórico para embasamento desta monografia, a seção seguinte apresentará trabalhos relacionados à este.

## 3 TRABALHOS RELACIONADOS

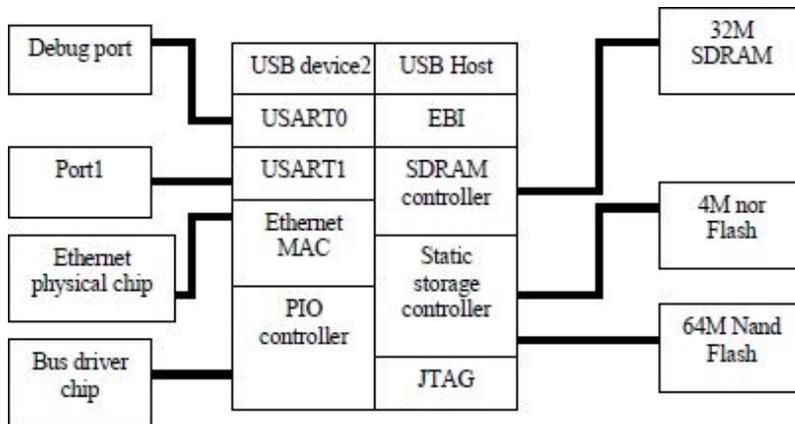
O capítulo 2 aborda os principais conceitos utilizados para a concepção do protótipo proposto. Neste capítulo há a análise de alguns trabalhos publicados semelhantes a este, como também um comparativo entre eles e suas propostas apresentadas. Este comparativo tem como propósito avaliar a viabilidade técnica sob o que está sendo proposto nesta monografia, analisando os resultados decorrentes nestes trabalhos observados estabelecendo uma relação com os resultados obtidos no desenvolvimento da *shield* proposta.

### 3.1 Estudo 1

O estudo 1, *ARM9-based Control System for LED Large Screen Display*, foi realizado por Qin Song e Yan Sun em 2010 na China. O artigo descreve um sistema de controle para um *Display* de LED baseado numa arquitetura ARM9 em que o sistema utiliza um microprocessador AT91RM9200 de 32 *bits* para o controle do *display*. Os autores realizaram comparativos deste microprocessador com microcontroladores, chegando a conclusão que com o sistema desenvolvido o microprocessador oferece melhorias no processamento, arquitetura e escopo de endereçamento como também apresenta uma *interface* externa. Para simplificar o projeto do circuito foi utilizada uma *interface* GPIO com AT91RM9200 diretamente implementada para executar a unidades lógicas do *display* LED.

A figura 08 ilustra a estrutura do *hardware* do sistema desenvolvido por Qin Song e Yan Sun.

Figura 08: Estrutura do *hardware*.



Fonte: Song e Sun, 2010.

### 3.2 Estudo 2

*Design and Development of Atmega 2560 AVR Microcontroller Based Control and Monitoring System of a Two Jaw Slit*, é o título do estudo 2 desenvolvido na Índia em 2010 pelos Suman Kumar Guha, Sumantra Bhattacharya, P. Y. Nabhiraj, Chinmay Nandi. Os autores projetaram o *K-500 Superconducting Cyclotron of Variable Energy Cyclotron Centre (VECC)*, com o objetivo de acelerar íons positivos pesados e leves em divergentes níveis de energia, a figura 09 ilustra o conjunto de fendas que fazem parte da estrutura física do sistema. Para o controle, tanto local como remoto, e para o monitoramento do VECC foi utilizado um microcontrolador ATmega2560.

Figura 09: Conjunto de fendas do VECC.



Fonte: Guha *et al*, 2010.

### 3.3 Estudo 3

Com o apoio da empresa Texas Instruments e da NASA, Gang Sun, Joseph Morgan, Jay Porter e Hester Yim em 2015 desenvolveram uma nova plataforma integrada modular, possibilitando ser “empilhada”, voltada para o ambiente educacional. O estudo publicado, *Development of a Novel Modular Integrated Stackable Layer - Analog System Environment (MISL - ASE) Platform for Embedded Systems Education*, descreve que o objetivo do projeto era fornecer um ambiente de *hardware* educacional abrangente para três cursos de *design* de sistema embarcado e dois cursos de *design* de ponta no Programa de Tecnologia de Engenharia de Sistemas Eletrônicos (ESET) na Texas A&M University.

Possuindo um microcontrolador TI MSP430 a plataforma MISL-ASE tem uma *interface* direta com a placa ASE, englobando vários periféricos digitais e analógicos, tais como GPIOs de entrada/saída, LEDs, *display* de 7 segmentos, conversores digitais e analógicos para simulações, entre outros ilustrados na figura 10. Estão disponíveis também algumas *interfaces* e protocolos de comunicação (UART, SPI, I2C).

Figura 10: Exemplos de periféricos da plataforma MISL-ASE.



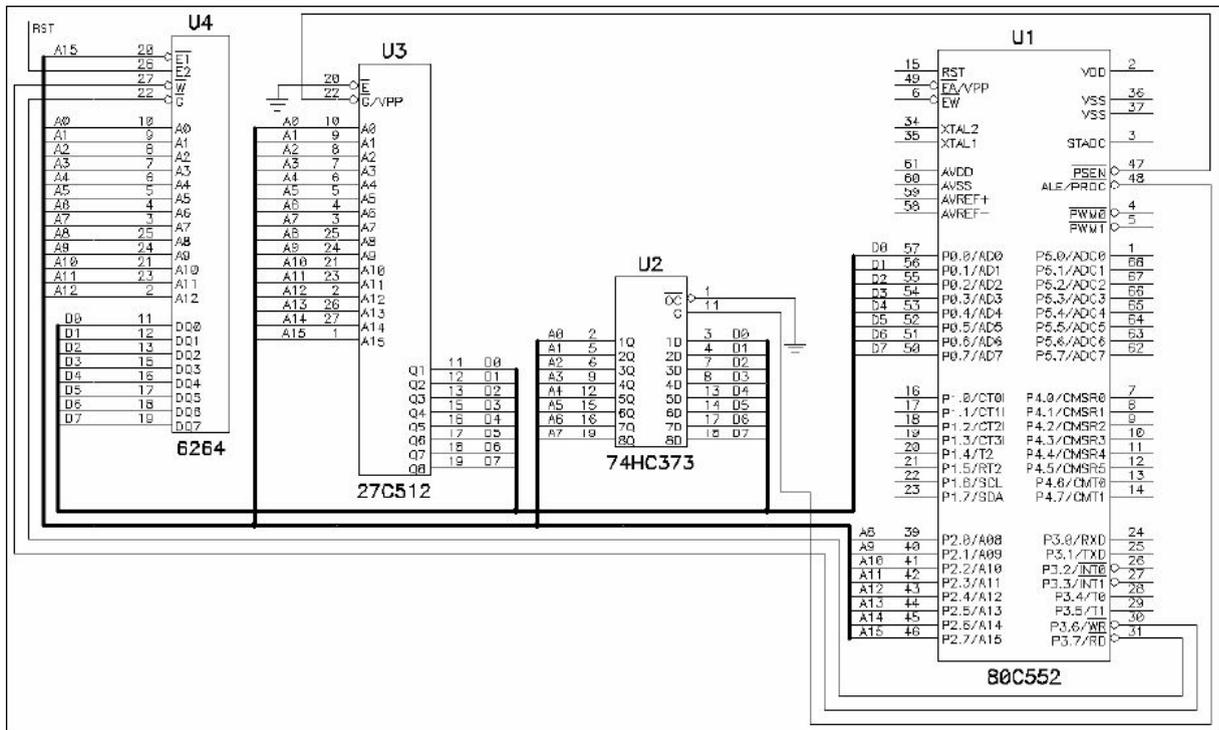
Fonte: Sun *et al*, 2015.

### 3.4 Estudo 4

O estudo 4 foi desenvolvido em 2002 na Universidade Regional de Blumenau, por Eduardo Klaus Barg. O trabalho especifica e implementa um protótipo de *hardware* que utiliza a arquitetura em 8 *bits* de um microcontrolador com o objetivo de executar o controle de temperatura de um ambiente. Barg em seu protótipo utilizou o MCU 80C552 da Philips por ele apresentar entradas e saídas analógicas e PWM *on-chip*.

O microcontrolador 80C552 não possui internamente uma memória de programa (ROM), portanto foi necessário a utilização de uma memória externa (EPROM). O autor afirma que superdimensionou a memória 27C512 de 64 Kbytes para o projeto, mas justifica esclarecendo que foi para dar mais versatilidade ao protótipo, optando também por incluir uma memória RAM externa. A figura 11 ilustra o esquemático eletrônico das conexões utilizadas entre o microcontrolador e as memórias externas.

Figura 11: Conexão do microcontrolador e as memórias EPROM e RAM.



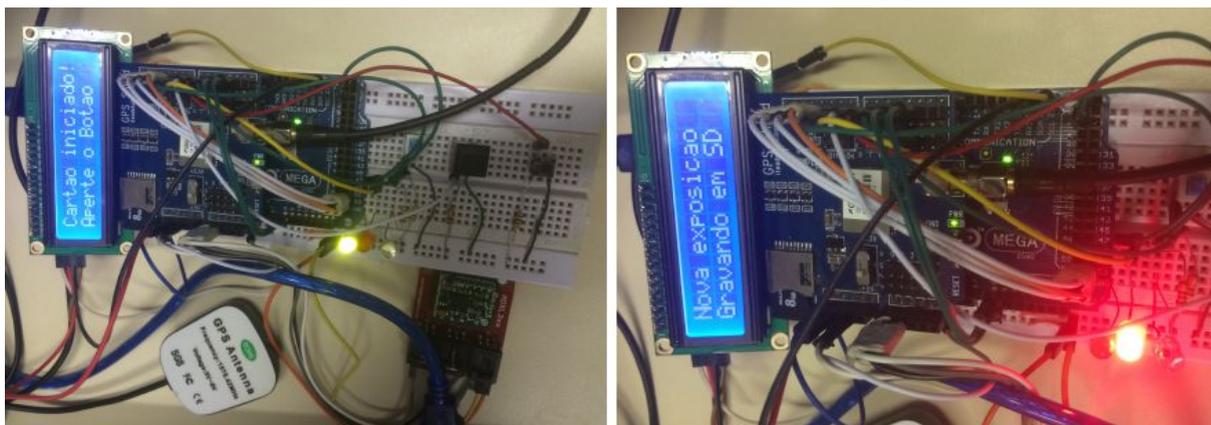
Fonte: Barg, 2002.

### 3.5 Estudo 5

Filipe Emídio Tôrres em 2015 para seu trabalho, processamento digital de sinais de vibração adquiridos com *hardware* dedicado e visualização científica com localização geográfica, avaliou dispositivos eletrônicos comerciais que atuavam em transporte viário. Em seu estudo verificou que alguns destes equipamento comerciais apresentavam limitações em seu *hardware*, com isso decidiu desenvolver um novo aparelho para efetuar tais medição, análise e visualização dos sinais. Para a captura dos sinais, Tôrres utilizou a placa embargada Arduino Mega 2560, e para o processamento foi utilizado uma Raspberry PI.

O autor juntamente com uma empresa de viação realizou a identificação dos problemas relacionados aos métodos comerciais, concluindo que tais sistemas apresentam um custo muito elevado e são limitados. Para sanar tais problemas foi desenvolvido um sistema que utiliza uma placa Arduino Mega 2560, sobre ela encaixado um módulo GPS/Cartão de Memória e ainda uma placa Raspberry PI conectada na porta serial para o recebimento dos dados adquiridos. A figura 12 ilustra a montagem deste *hardware*.

Figura 12: Sistema sendo mostrado em dois estados.



Fonte: Tôrres, 2015.

### 3.6 Comparativo entre os Trabalhos Relacionados

O quadro 01 apresenta um comparativo realizado entre os trabalhos apresentados neste capítulo. Principais características analisadas e comparadas foram: utilização de microcontrolador; utilização de memória externa; *hardware* dedicado e desenvolvimento de *hardware*.

Quadro 01: Comparativos dos estudos relacionados.

	Estudo 1 Song e Sun, 2010.	Estudo 2 Guha et al, 2010	Estudo 3 Sun et al, 2015.	Estudo 4 Barg, 2002.	Estudo 5 Tôres, 2015.
Microcontrolador	Utilizam um microprocessador AT91RM9200	Microcontrolador ATmega2560.	Microcontrolador TI MSP430	Microcontrolador 80C552	Microcontrolador ATmega2560.
Memória Externa	Utilizaram	Não utilizaram	Não utilizaram	Utiliza	Não utiliza
<i>Hardware</i> dedicado	Utilizaram	Não utilizaram	Utilizaram	Não utiliza	Utiliza
Desenvolvimento de <i>hardware</i>	Desenvolveram o <i>hardware</i>	Não desenvolveram	Desenvolveram o <i>hardware</i>	Desenvolveu o <i>hardware</i>	Desenvolveu o <i>hardware</i>

Fonte: Autor, 2020.

Cada estudo denota uma aplicabilidade diferente, mas apresentam algumas características comuns. Observa-se que tanto no Estudo 1 como no Estudo 4 houve a utilização de memória externa pois houve a necessidade nos projetos desenvolvidos, da mesma forma que os Estudos 1, 3 e 5 apresentam a utilização de *hardware* dedicado, Já os Estudos 2 e 5 utilizaram o microcontrolador ATmega2560, o mesmo deste projeto.

Neste capítulo foram comparados estudos que apresentam propostas similares ao presente trabalho, no capítulo 4 será apresentado os métodos utilizados para o desenvolvimento deste.

## 4 MATERIAIS E MÉTODOS

Este capítulo abordará os procedimentos metodológicos que conduziram a concepção deste trabalho, estabelecendo os métodos científicos, ferramentas de abordagem e metodologias utilizadas.

### 4.1 Pesquisa científica

Silveira e Córdova (2009) afirmam “A pesquisa é a atividade nuclear da Ciência”. Eles referem-se à pesquisa como uma inquisição, um procedimento meticuloso e intensivo que objetiva interpretar e descobrir o fatos de um determinada realidade.

#### 4.1.1 Abordagem

O presente trabalho tem caráter qualitativo e quantitativo. De cunho qualitativo a pesquisa apresenta análises e interpretações, utilizando estas análises e interpretações o método quantitativo deduz de forma lógica atributos mensuráveis, sendo apresentados como resultados. O método quantitativo foi utilizado para realizar as análises dos tempos obtidos no processo de inicialização do *display* LCD.

#### **4.1.2 Natureza**

A pesquisa, quanto a sua natureza, é classificada como aplicada, pois tem como objetivo fornecer conhecimentos para aplicações práticas, voltado para soluções de adversidades específicas. Neste sentido, foi desenvolvida uma *shield* compatível com a plataforma Arduino Mega 2560, que possibilita utilizar os pinos com recursos de XMEM de forma a verificar a otimização na comunicação com dispositivos externos.

#### **4.1.3 Objetivos**

Com objetivo exploratório, a pesquisa proporciona um entendimento do problema, otimização de *hardware* e *software*, tornando-o mais claro e explorando hipóteses de melhoria no desempenho do microcontrolador.

#### **4.1.4 Procedimentos**

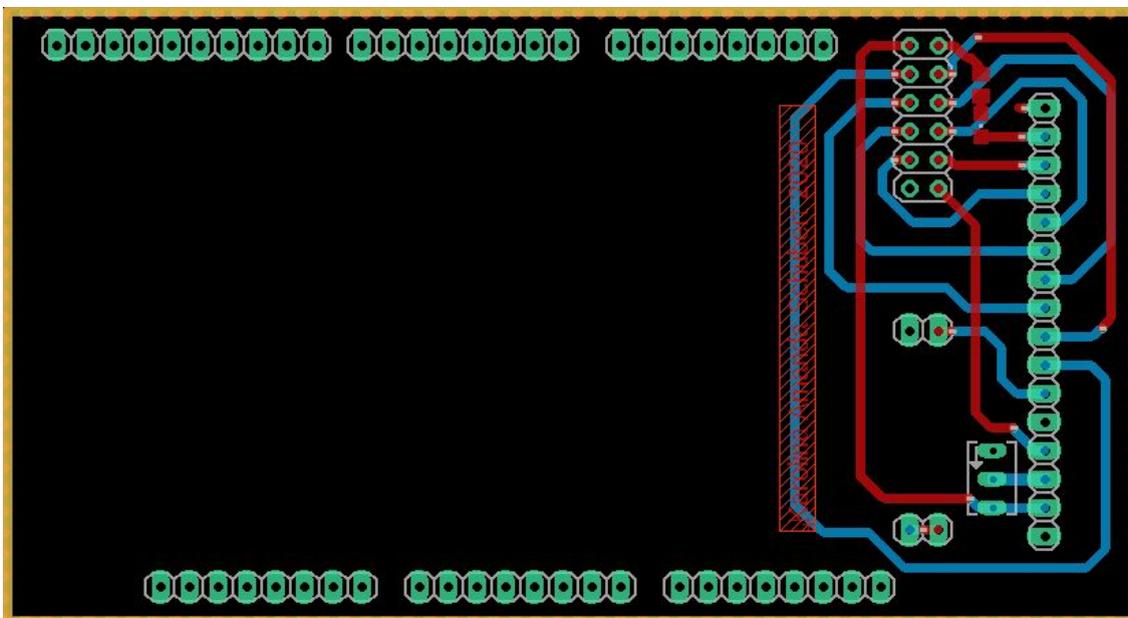
De caráter experimental, a pesquisa apresenta como objeto de estudo a otimização na forma de comunicação entre microcontrolador ATmega2560 e as *shields* que apresentam *display* LCD. Foram definidas oito formas de análises que resultaram em diferentes tempos na forma de comunicação entre as placas.

#### **4.2 Tecnologias**

Os tópicos a seguir descrevem as tecnologias empregadas no desenvolvimento da *shield* como também os métodos utilizados para as análises das placas, comercial e desenvolvida que apresentam o *display* LCD e acoplam-se na placa embarcada Arduino Mega 2560.



Figura 14: Board da shield LCD.

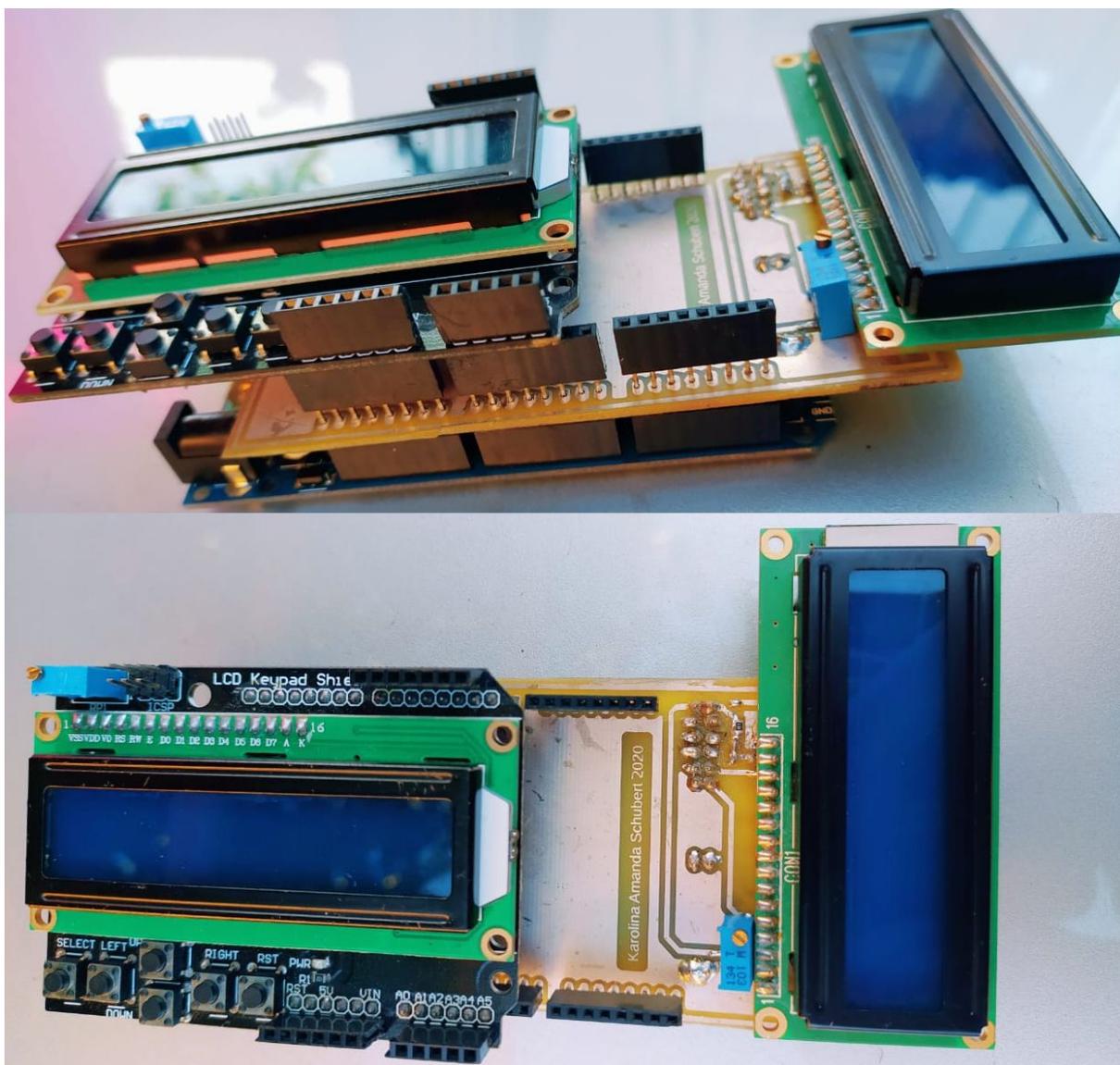


Fonte: Autor, 2020.

Para o desenvolvimento da placa em circuito impresso foi utilizado a sala 412-anexa, conhecida como “oficina”, laboratório presente no prédio 11 do campus da Univates. Após ser prensado na placa de fenolite o esquemático, tanto da parte inferior (*bottom*) como a parte superior (*top*), a placa foi corroída em percloroeto de ferro, furada para que os componentes pudessem ser encaixados, envernizada evitando assim a oxidação e concluída com a solda dos respectivos componentes.

A placa desenvolvida manteve os pinos genéricos do Arduino Mega 2560 disponíveis para que pudessem ser utilizados com outras *shields* ou até mesmo em conexões com *proto-board*. A figura 15 ilustra a placa *shield* LCD finalizada acoplada no Arduino Mega 2560 e sobre ela a *shield* LCD Keypad Shield.

Figura 15: *Shield* LCD.



Fonte: Autor, 2020.

#### 4.2.2 LCD Keypad Shield

O *display* pode ser encontrado nas mais diversas aplicações, com o intuito de facilitar a ligação física dos *displays* às placas Arduino a empresa DFROBOT comercializa uma *shield* com *display* LCD e 5 botões multipropósitos que são conectados apenas a uma entrada analógica. A figura 16 ilustra a *LCD Keypad Shield*.

Figura 16: *LCD Keypad Shield*.



Fonte: DFROBOT, sem ano.

A *LCD Keypad Shield* possui as seguintes características:

- LCD com luz de fundo azul e caracteres brancos;
- *Display* com duas linhas e dezesseis colunas;
- Seis botões tácteis;
- Pinos de entradas e saídas;
- Um canal ADC;
- Potênciômetro para o ajuste de *backlight*.

#### 4.2.3 *Display* LCD

O modelo *display* LCD presente em ambas *shields* (desenvolvida e a *LCD Keypad Shield*) possibilita uma *interface* visual podendo indicar a tarefa que o microcontrolador está executando, mostrando na tela textos, números e símbolos. Os modelos mais vendidos de *displays* LCDs possuem a dimensão matricial de 16 caracteres por 2 linhas (16X2).

Presente na placa do *display* LCD há também um chip responsável pelo seu controle. O chip HD44780 é desenvolvido pela empresa Hitachi, ele possibilita dois modos distintos de comunicação dos dados impressos na tela, modo 4 *bits* e o modo 8 *bits*. Ambos os modos são configurados através do código de programação para a

utilização do *display* LCD. Estes modos são referentes à ligação dos barramentos do *display*, no modo 4 *bits* são utilizados apenas quatro dos oito pinos de dados da placa, já na operação de 8 *bits* todos os pinos são utilizados.

A *shield* LCD Keypad *Shield* faz o uso do modo de 4 *bits*, já a *shield* desenvolvida utiliza o método de operação de 8 *bits*. Optou-se por utilizar o modo de 8 *bits* na *shield* desenvolvida pois a comunicação dos dados pelo barramento ocorre através de oito pinos de comunicação comunicando assim 1 *byte*, reduzindo o uso da memória, fisicamente utilizam-se mais pinos mas há um ganho e otimização no programa. A *shield* comercial, como já citado, opera pelo modo de 4 *bits*, ocupando quatro pinos do microcontrolador, sua transmissão de dados neste modo é feita em dois pacotes, são transferidos 4 *bits* pelo barramento de comunicação, resultando um número menor de pinos utilizados, entretanto gerando a desvantagem em um uso maior da memória. O quadro 02 apresenta os pinos de conexão do *display* LCD.

Quadro 02: Pinagem *display* LCD.

Pino	Descrição
1	Terra (GND)
2	5 Vcc
3	Ajuste de contraste (VO)
4	Seleção de registro (RS). Comando = 0; Dados = 1
5	Leitura / gravação (R / W). Write = 0; Read = 1
6	Relógio ( <i>Enable</i> ). Borda descendente acionada
7	Bit 0 (não usado na operação de 4 bits)
8	Bit 1 (não usado na operação de 4 bits)
9	Bit 2 (não usado na operação de 4 bits)
10	Bit 3 (não usado na operação de 4 bits)
11	Bit 4
12	Bit 5
13	Bit 6
14	Bit 7
15	Ânodo da luz de fundo
16	Catodo de luz de fundo

Fonte: Autor, 2020.

#### 4.2.4 Software

Com o intuito de realizar mais análises comparativas foram desenvolvidos programas utilizando a IDE do Arduino e separados em três categorias, a primeira

utiliza a biblioteca LiquidCrystal, o uso desta biblioteca, além de facilitar a programação, já configura a programação em baixo nível das configurações de barramento entre as placas Arduino e o *display* LCD. Ela possibilita iniciar o *hardware* do *display* LCD com os barramentos de 8 *bits*, que foi utilizado com a *shield* desenvolvida, como também o modo de 4 *bits*, utilizado com a *shield* comercial.

A segunda foi elaborada em linguagem C++ utilizando diretamente endereços do microcontrolador, e por terceiro, também desenvolvido em linguagem C++, utiliza o recurso XMEM do microcontrolador, utilizando os registradores em mapeamento da memória externa do microcontrolador. Todos os programas imprimem na tela do *display* LCD o texto “*Hello, world!*”, como também ao iniciar o componente elas executam as funções de inicialização, pois como citado no tópico 4.3 ao iniciar o *display* LCD ele necessita ser configurado, via *software*.

#### **4.2.5 Verificação do tempo de inicialização do *hardware***

Para possibilitar a medição do tempo em que o *hardware* utiliza para inicializar o *display* LCD foi incluído ao código principal comandos que ativam e desativam a saída de um pino do Arduino Mega 2560. O pino é ativado antes da configuração do *display* LCD iniciar, após configurado o *hardware* o pino é desativado. A medição do tempo é realizada por um osciloscópio digital, medindo o intervalo de um pulso de nível alto para nível baixo.

Foram realizadas oito análises distintas, nas quais as duas primeiras foram realizadas com a LCD Keypad Shield, o anexo 03 apresenta o esquemático da placa. A primeira análise, denominada como “A” utiliza a biblioteca LiquidCrystal e para a segunda análise, “B”, foi utilizado o código em linguagem C++ utilizando diretamente os endereços do microcontrolador. Para ambos ensaios a ligação física ocorreu através do acoplamento da shield com o Arduino Mega 2560, os principais pinos da placa embarcada utilizados pelo LCD Keypad Shield são apresentados no quadro 03.

Quadro 03: Ligação entre o Arduino Mega 2560 e a LCD Keypad shield.

Arduino Mega 2560			<i>Display LCD</i>	
<i>PORT</i>	<i>bit</i>	Pino físico	Pino físico	Descrição
PORTB	4	10	15	A
PORTE	3	5	12	B5
PORTG	5	4	11	B4
PORTH	3	6	13	B6
PORTH	4	7	14	B7
PORTH	5	8	4	RS
PORTH	6	9	6	E

Fonte: Autor, 2020.

Com a *shield* LCD desenvolvida foram realizadas seis análises, intituladas de “C.1”, “C.2”, “D.1”, “D.2” e “E”. Tanto os ensaios “C” e “D” seguem a mesma proposta dos ensaios “A” e “B” respectivamente, tendo o diferencial em que “C.1” e “D.1” utilizam o modo de comunicação do barramento em 4 *bits* do *display* LCD e “C.2” e “D.2” utilizam o modo de comunicação do barramento em 8 *bits* do *display* LCD. Primeiramente, nas análises “C”, utilizando a biblioteca LiquidCrystal e por segundo, nas análises “D”, utilizando diretamente os endereços do microcontrolador. A análise “E” utiliza o recurso XMEM do microcontrolador com o modo de comunicação do barramento em 8 *bits* do *display* LCD, o quadro 04 descreve os pinos referente às ligações que o *display* LCD presente na placa desenvolvida faz quando esta é acoplada ao Arduino Mega 2560.

Quadro 04: Ligação entre o Arduino Mega 2560 e a *shield* desenvolvida.

Arduino Mega 2560		Display LCD		
PORT	bit	Pino físico	Pino físico	Descrição
		GND	1	GND
		VCC	2	5 VCC
		Potenciometro	3	V0
PORTC	bit 6	31	4	RS
		GND	5	R/W
PORTG	bit 0	41	6	E
PORTA	bit 0	22	7	B0
PORTA	bit 1	23	8	B1
PORTA	bit 2	24	9	B2
PORTA	bit 3	25	10	B3
PORTA	bit 4	26	11	B4
PORTA	bit 5	27	12	B5
PORTA	bit 6	28	13	B6
PORTA	bit 7	29	14	B7
		Resistor 330 $\Omega$	15	A
		GND	16	K

Fonte: Autor, 2020.

Neste capítulo foram relatadas as atividades realizadas para a elaboração da *shield* como também as análises dividiram-se, o próximo capítulo apresentará os resultados obtidos através destas análises.

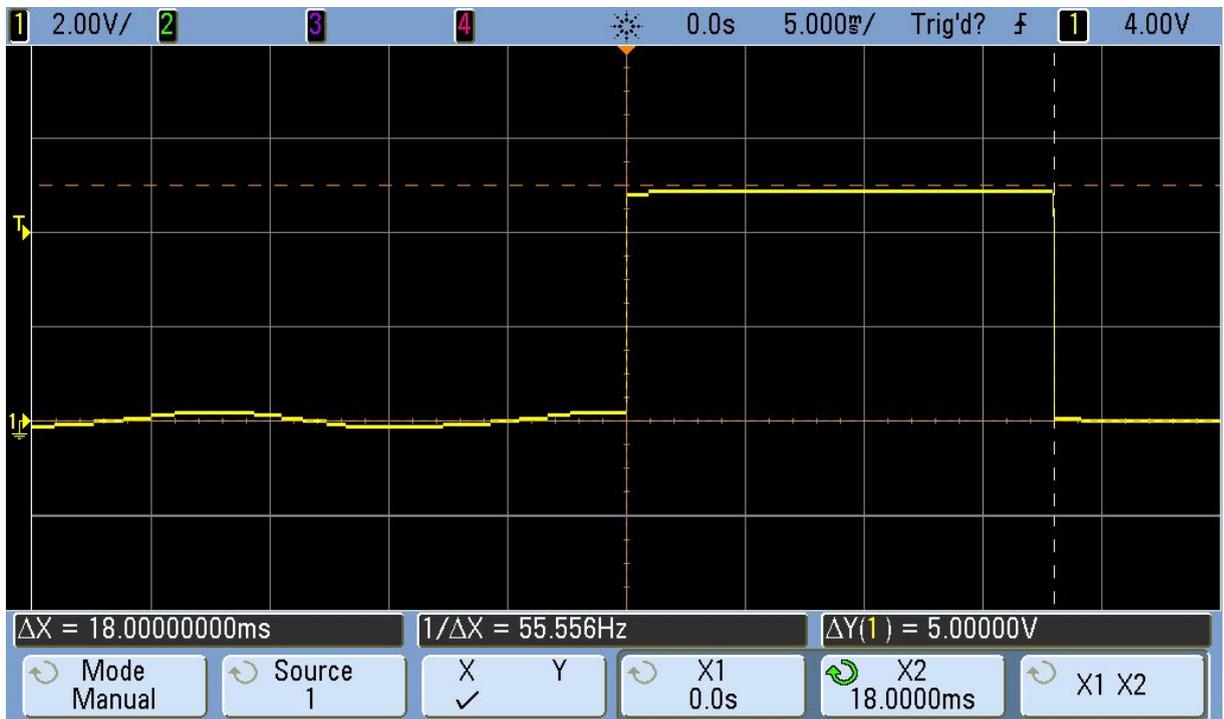
## 5 RESULTADOS

Neste capítulo serão apresentados os tempos em que o *display* LCD necessita para ser inicializado, oriundos dos testes descritos no capítulo anterior. Os teste foram realizados diversas vezes, observando que os resultados sempre mantiveram os mesmos valores. O tempo é obtido a partir do delta entre X1 e X2, em que X1 se encontra quando o pino é ativado e X2 está situando quando quando o pino é desativado.

### 5.1 Teste utilizando a LCD *Keypad Shield*

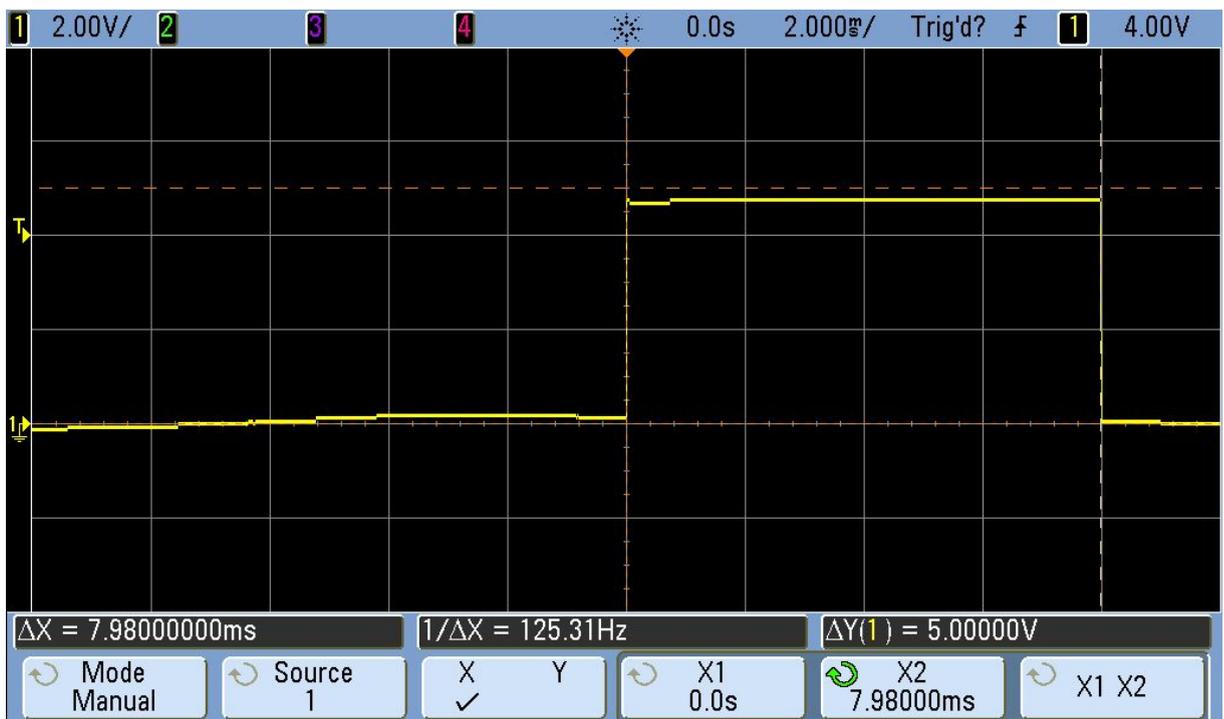
A seção 5.1 apresenta os resultados provenientes das medições realizadas com o osciloscópio dos testes realizados com a placa comercial LCD *Keypad Shield*. Ilustrado na figura 17 tem-se a medição do tempo da primeira análise, denominada como “A”, utilizando o *hardware* da placa comercial LCD *Keypad Shield* e na parte de *software* utilizando a biblioteca LiquidCrystal. A figura 18 apresenta a medição da análise “B”, mesmo *hardware* da análise “A”, porém com *software* desenvolvido em linguagem C++ utilizando os registradores do microcontrolador.

Figura 17: Medição da análise "A".



Fonte: Autor, 2020.

Figura 18: Medição da análise "B".

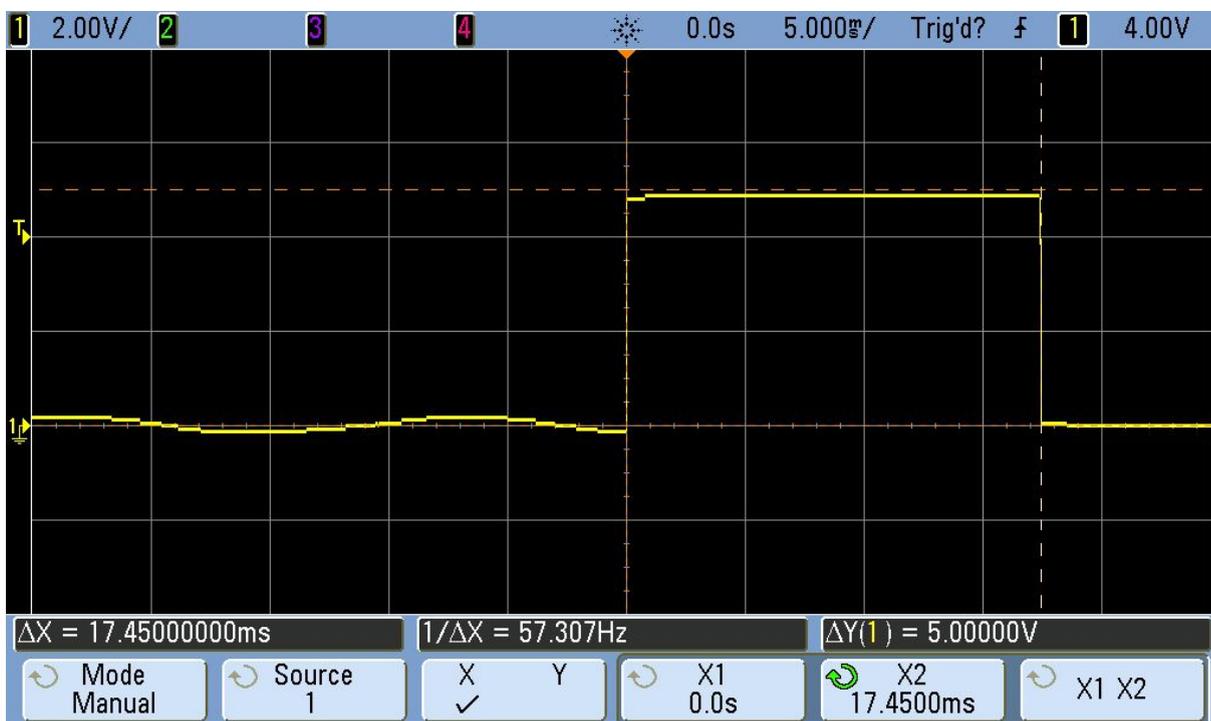


Fonte: Autor, 2020.

## 5.2 Teste utilizando a *shield* desenvolvida

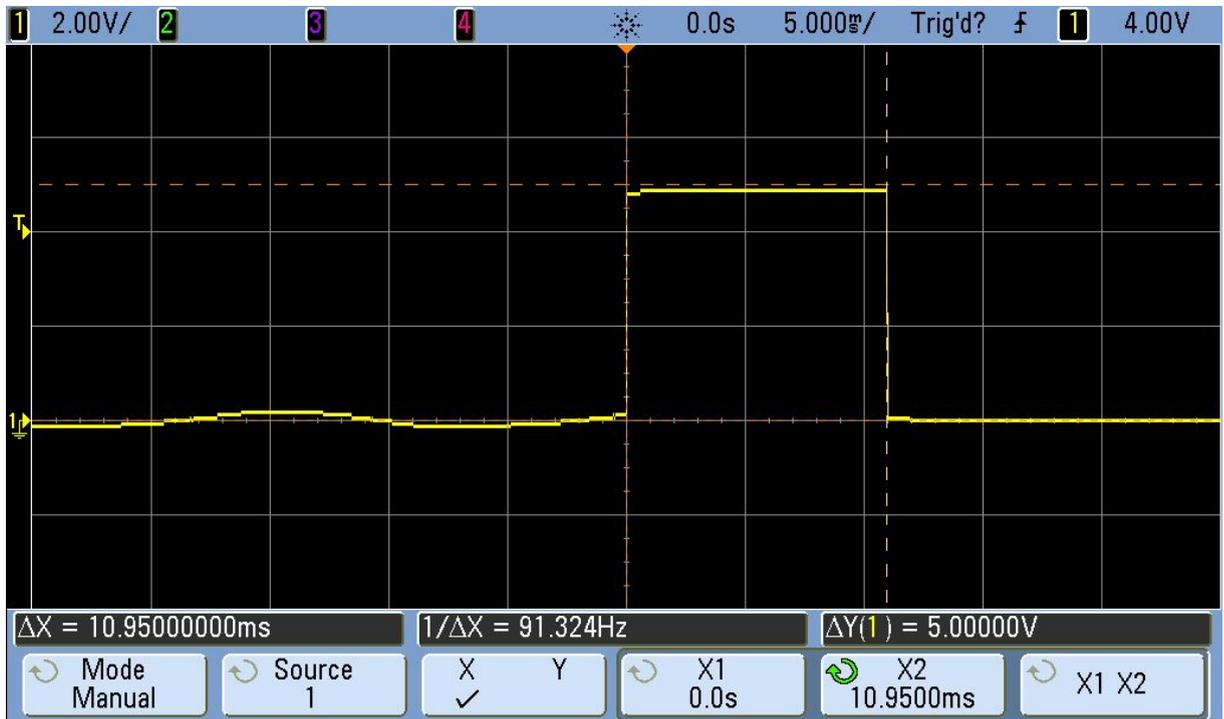
A seção 5.2 apresenta os resultados provenientes das medições realizadas com o osciloscópio dos testes realizados com a *shield* desenvolvida. As figuras 19 e 20 ilustram as análises “C.1” e “C.2” em que o *software* utiliza a biblioteca LiquidCrystal, as figuras 21 e 22 são relativas às análises “D.1” e “D.2” que executa o *software* com os endereços do microcontrolador, já a figura 23 é correspondente a análise “E”, que utiliza os registros em mapeamento em memória.

Figura 19: Medição da análise “C.1”.



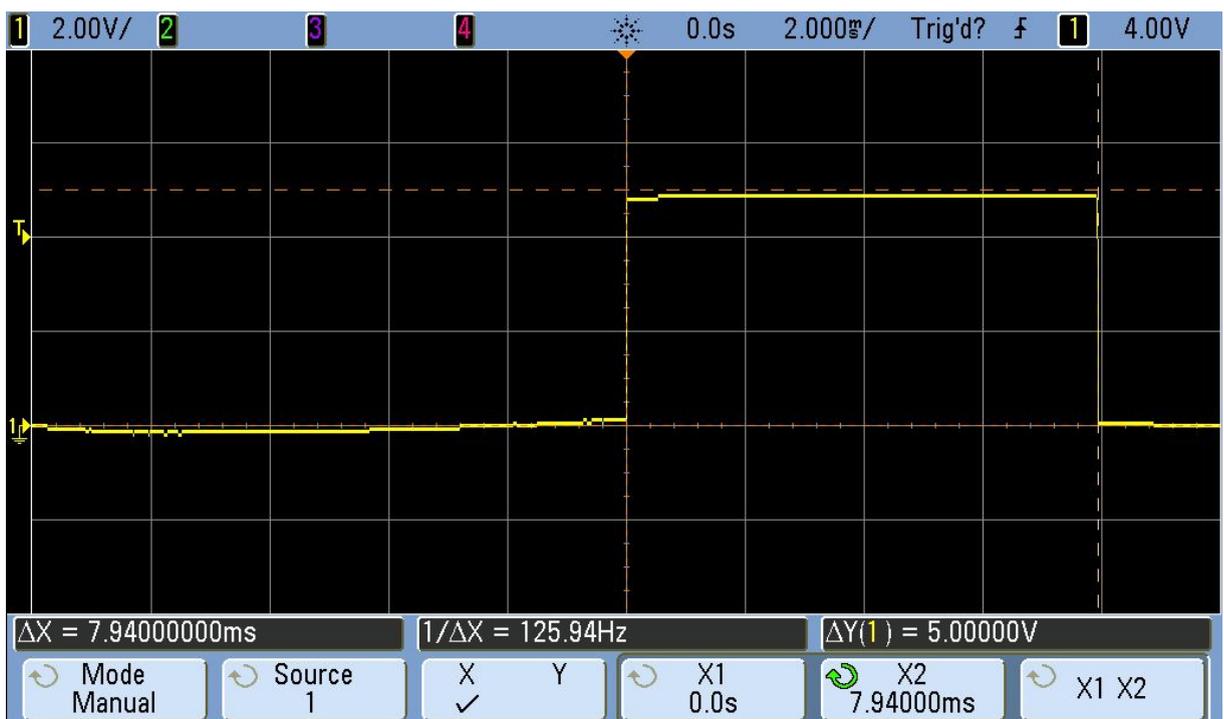
Fonte: Autor, 2020.

Figura 20: Medição da análise "C.2".



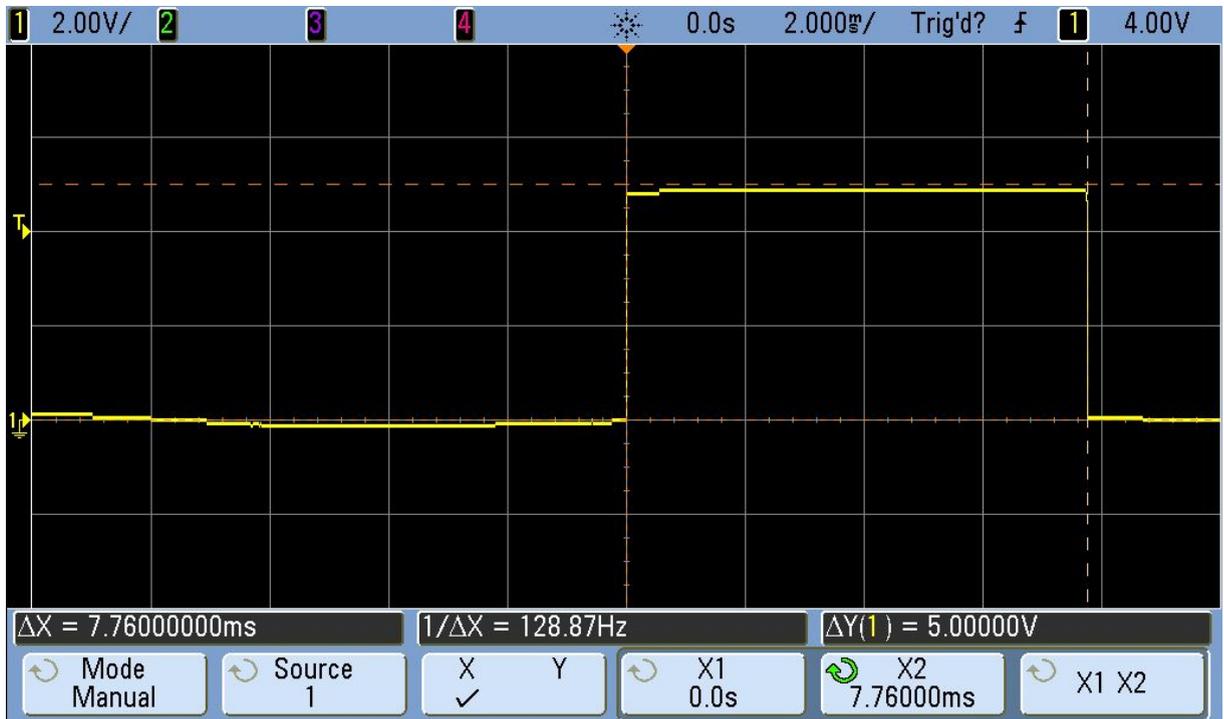
Fonte: Autor, 2020.

Figura 21: Medição da análise "D.1".



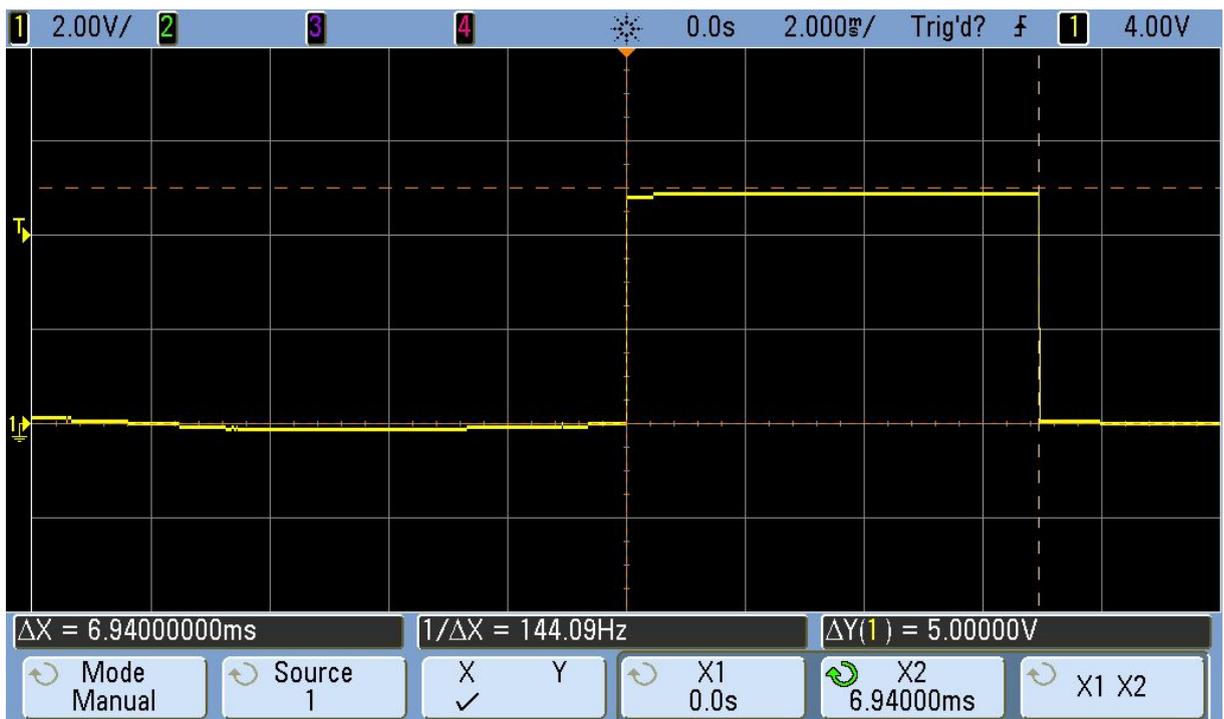
Fonte: Autor, 2020.

Figura 22: Medição da análise "D.2".



Fonte: Autor, 2020.

Figura 23: Medição da análise "E".



Fonte: Autor, 2020.

### 5.3 Comparação dos resultados

O quadro 05 apresenta os resultados oriundos das medições realizadas com o osciloscópio. Percebe-se que na análise “A” o tempo medido foi de 18 ms, esta análise foi realizada com o *hardware* da *shield* LCD Keypad Shield e o *software* desenvolvido com a biblioteca LiquidCrystal. Ainda com o mesmo *hardware* dispõem-se a análise “B” em que o tempo medido foi de 7,98 ms, esta diferença entre as análises deu-se pelo *software* em que na segunda análise fora desenvolvido em linguagem C++, acessando diretamente os endereços do microcontrolador.

Utilizando a placa desenvolvida e com três programas discrepantes obtiveram-se cinco resultados diferentes. As análises “C.1” (utilizando barramento de comunicação de 4 *bits* com o *display* LCD) e “C.2” (utilizando barramento de comunicação de 8 *bits* com o *display* LCD) foram desenvolvidas com o *software* utilizando a biblioteca LiquidCrystal, em que apresentaram o tempo de 17,45 ms e 10,95 ms respectivamente.

Utilizando os endereços do microcontrolador na programação do *software* foi possível executar as análises “D.1” que apresentou 7,94 ms e “D.2” com 7,76 ms, em que “D.1” utilizou barramento de comunicação de 4 *bits* com o *display* LCD e “D.2” utilizou o barramento de 8 *bits*. Por último, a análise “E”, em seu programa estava habilitado a XMEM, possibilitando o uso dos registros de mapeamento em memória, nesta análise obteve-se o melhor tempo, 6,94 ms.

Quadro 05: Comparação dos testes realizados.

Análises	Tempo	Tamanho do código compilado
A	18 ms	2166 bytes
B	7,98 ms	600 bytes
C.1	17,45 ms	2166 bytes
C.2	10,95 ms	2176 bytes
D.1	7,94 ms	544 bytes
D.2	7,76 ms	600 bytes
E	6,94 ms	484 bytes

Fonte: Autor, 2020.

Confrontando as análises “A” e “B”, em que o diferencial está presente apenas no *software*, visto que melhorias no *software* acarretam na diminuição do código compilado, gera um ganho de 55,67% no tempo de comunicação na inicialização do *display* LCD. Com a otimização tanto no *software* como também no *hardware*, compara-se a análise “A” com a “E” resultando uma melhoria de 61,45%.

## 6 CONCLUSÃO

Este trabalho propôs a análise de desempenho tanto de *hardware* como de *software* para uma plataforma computacional microcontrolada através da utilização de registradores da memória externa. Foram comparadas duas placas, *shields*, distintas em que uma delas é comercializada e a outra foi desenvolvida para este trabalho. A LCD *Keypad Shield*, com seu *hardware*, possibilitou o desenvolvimento de dois códigos, em que foram realizadas as análises “A” e “B”, a *shield* que foi projetada para que houvesse uma otimização, possibilitando o acesso aos registradores da XMEM, viabilizou a realização de cinco análises com diferentes códigos.

Ao término das análises foi possível concluir que, com as melhorias na estrutura do *hardware*, ao desenvolver a *shield*, foi possível aprimorar o *software*, utilizando o mapeamento em memória externa do microcontrolador, gerando um ganho de 61,45 % na inicialização do *display* LCD. Pode-se afirmar que todos os objetivos do presente trabalho foram atendidos com sucesso.

Como sugestão para trabalhos futuros sugere-se a utilização da *interface* de memória externa com outros periféricos, pois como constatado, o envio de dados ocorreu de forma mais rápida com este método ao utilizar o microcontrolador.

## REFERÊNCIAS

ARDUINO. **Arduino**. 2020. <https://www.arduino.cc/>. Acesso em: 05 de mar. 2020.

ATMEL. **Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V**. Atmel Corporation, 2014. 435p. Disponível em: <[http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561\\_datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf)>. Acesso em: 18 de mar. 2020.

AURELIANO, Andre. **Microcontroladores**. Fiozera. 04 de abril de 2017. Disponível em: <<https://fiozera.com.br/microcontroladores-914a59cbf7de>>. Acesso em: 18 de mar. 2020.

BARG, Eduardo Klaus. **PROTÓTIPO DE UM CONTROLADOR DE TEMPERATURA BASEADO EM LÓGICA FUZZY UTILIZANDO UM MICROCONTROLADOR**. Universidade Regional de Blumenau, Centro de Ciência Exatas e Naturais, Curso de Ciências da Computação. Blumenau, 2002. Acesso em: 18 de agos. 2020.

CAMPOS, Augusto. **LCD: Arduino e o display LCD 1602 (HD44780, SPLC780D)**. BR-Arduino.org, Do LED ao Arduino: aprendendo eletrônica no século 21, 22/12/2014. Disponível em: <<https://br-arduino.org/2014/12/lcd-arduino-e-o-display-lcd-1602-hd44780-splc780d.html>>. Acesso em: 20 de mar. 2020.

CORRÊA, Ricardo Varela. **OTIMIZAÇÃO DE DESEMPENHO UTILIZANDO CONTADORES DE HARDWARE**. – São José dos Campos: INPE, 2020. 193p.

DFROBOT. **Gravity: 1602 LCD Keypad Shield For Arduino**. sem ano. Disponível em: <[https://wiki.dfrobot.com/LCD\\_KeyPad\\_Shield\\_For\\_Arduino\\_SKU\\_\\_DFR0009](https://wiki.dfrobot.com/LCD_KeyPad_Shield_For_Arduino_SKU__DFR0009)>. Acesso em: 06 de mar. 2020.

GUHA, Suman Kumar; BHATTACHARYA , Sumantra; NABHIRAJ, P. Y.; NANDI, Chinmay. **Design and Development of Atmega 2560 AVR Microcontroller Based Control and Monitoring System of a Two Jaw Slit**. 2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech). India. Aug. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8981123>>. Acesso em: 22 de ag. 2020.

GRFIM, Daniel; SILVA, Renan Tavares da; CRUZ, Willian Carlos Teixeira da. **Máquina de medir por coordenadas utilizando microcontrolador e sistema supervisorio**. 2018. 100 f. Trabalho de Conclusão de Curso (Superior de Tecnologia em Mecatrônica Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2018. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/11550>>. Acesso em: 09 de abr. 2020.

MEDEIROS, Savio Gomes Fernandes de. **SISTEMA DE MEDIÇÃO DE VARIÁVEIS ATMOSFÉRICAS USANDO MICROCONTROLADOR**. 2018. 58f. Universidade Federal Rural do Semi-Árido, Bacharelado em Ciência e Tecnologia. Campus Caraúbas. Caraúbas, Rio Grande do Norte. 2018. Disponível em: <[http://repositorio.ufersa.edu.br/bitstream/prefix/2906/2/SavioGFM\\_MONO.pdf](http://repositorio.ufersa.edu.br/bitstream/prefix/2906/2/SavioGFM_MONO.pdf)>. Acesso em: 09 de abr. 2020.

OLIVEIRA, João Carlos Silva de. **Paradigma da Orientação a Objetos na Linguagem de Programação C++**. ITEGAM-JETIA. Vol. 04, Nº 16, pp 116-128. December, 2018.

PATTERSON, David A.; HENNESSY, John L. **Computer Organization and Design: The Hardware/Software Interface**. 4th. ed. The Morgan Kaufmann series in computer architecture and design, 2011.

RAJEWSKI, Justin. **Memory Mapping**. 2018. Disponível em: <<https://alchitry.com/blogs/tutorials/memory-mapping>>. Acesso em: RASSOKHIN, Dmitrii. **The C++ programming language in cheminformatics and computational chemistry**. J Cheminform 12, 1º (2020). Disponível em : <<https://doi.org/10.1186/s13321-020-0415-y>>. Acesso em: 03 de maio, 2020.

SANTOS, Jean Willian; LARA JUNIOR, Renato Capelin de. **Sistema de automatização residencial de baixo custo controlado pelo microcontrolador ESP32 e monitorado via smartphone**. 2019. 46 f. Trabalho de Conclusão de Curso (Tecnologia em Automação Industrial) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/12133>>. Acesso em: 09 de abr. 2020.

SANTOS, Rodrigo Vieira dos. **USO DO ARDUÍNO E SHIELD ETHERNET PARA MONITORAMENTO DE LUMINOSIDADE, CONTROLE DE TEMPERATURA E DISPOSITIVOS**. Revista Científica Semana Acadêmica. Fortaleza, ano MMXVIII, Nº. 000129, 13/08/2018. Disponível em: <<https://semanaacademica.org.br/artigo/uso-do-arduino-e-shield-ethernet-para-monitoramento-de-luminosidade-controle-de-temperatura-e>>. Acessado em: 05 de mar. 2020.

SILVEIRA, Denise Tolfo; CÓRDOVA, Fernanda Peixoto. **Unidade 2–A pesquisa científica**. Métodos de pesquisa, v. 1, 2009. Disponível em: <[https://www.cesadufs.com.br/ORBI/public/uploadCatalogo/11315818082016Pratica\\_de\\_Pesquisa\\_I\\_Aula\\_2.pdf](https://www.cesadufs.com.br/ORBI/public/uploadCatalogo/11315818082016Pratica_de_Pesquisa_I_Aula_2.pdf)>. Acesso em: 29 out. 2020.

SONG, Qin; SUN, Yan. **ARM9-Based Control System for LED Large Screen Display**. 2010 Third International Conference on Intelligent Networks and Intelligent Systems. Shenyang, China, novembro 2010. Disponível em: <<https://ieeexplore.ieee.org/document/5693718>>. Acesso em: 22 de ag. 2020.

TANENBAUM, Andrew S. **Organização estruturada de computadores**. Andrew S. Tanenbaum, Todd Austin; tradução Daniel Vieira; revisão técnica Wagner Luiz Zucchi. - São Paulo: Pearson Prentice Hall, 2013.

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. Andrew S. Tanenbaum; tradução Ronaldo A. L. Gonçalves, Luíz A. Consularo, Luciana do Amaral Teixeira; revisão técnica Raphael Y. de Carmargo. -- 3. ed. -- São Paulo: Pearson Prentice Hall, 2009.

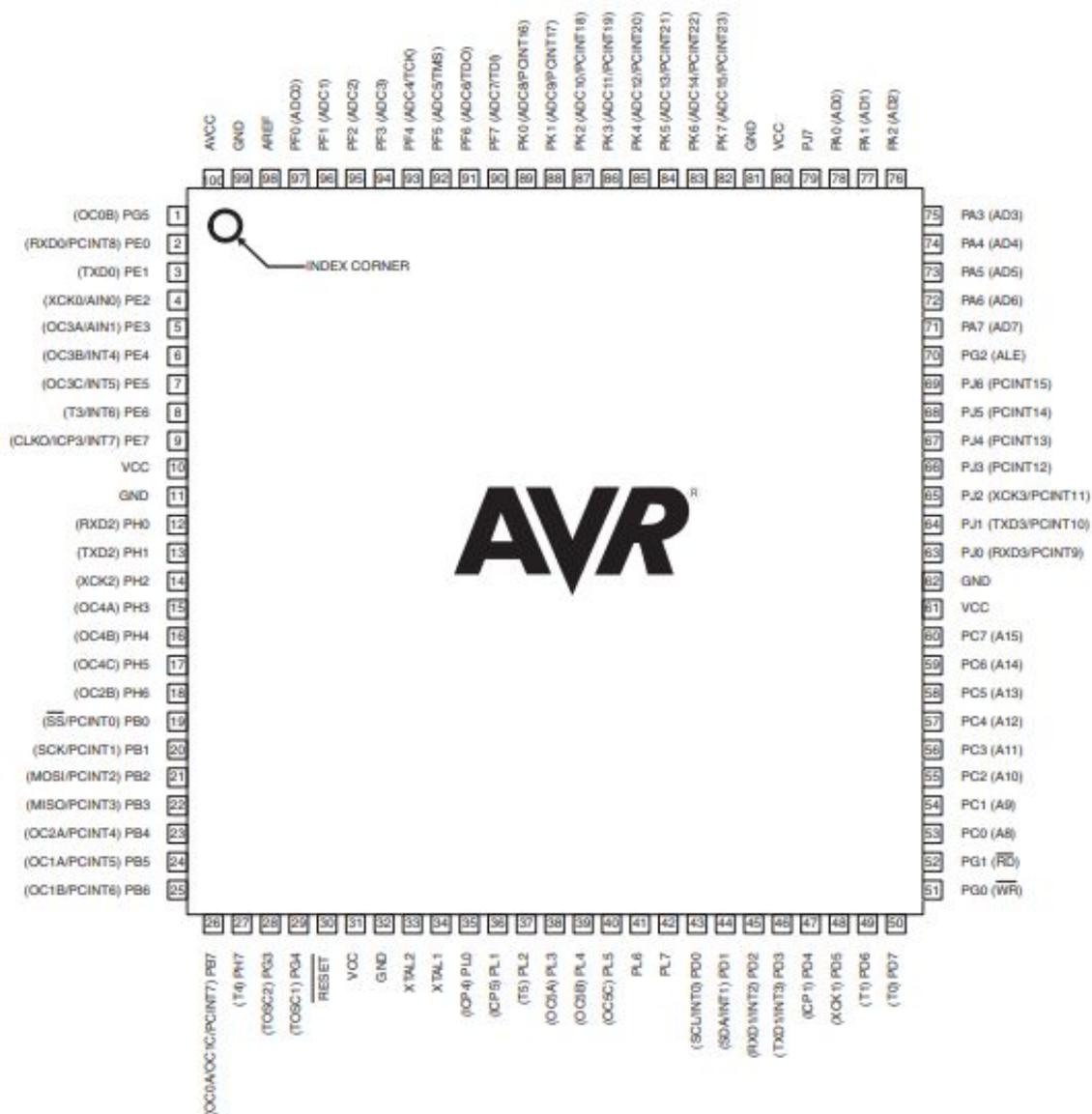
TÔRRES, Filipe Emídio. **Processamento digital de sinais de vibração adquiridos com hardware dedicado e visualização científica com localização geográfica**. Universidade de Brasília – UnB. Campus Gama – FGA. Brasília, DF - 2015. Acesso em: 22 de agos. 2020.

Visual Studio Code. **Microsoft(R) Visual Studio Code website**. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 02 de abr. 2020.

WIDMER, Neal S. **Sistemas digitais: princípios e aplicações**. Neals S Widmer, Gregory L. Moss, Ronald J. Tocci; [tradução Sérgio Nascimento]. - São Paulo: Pearson Education do Brasil, 2018.

## APÊNDICES E ANEXOS

### Anexo 01: *pinout* do microcontrolador



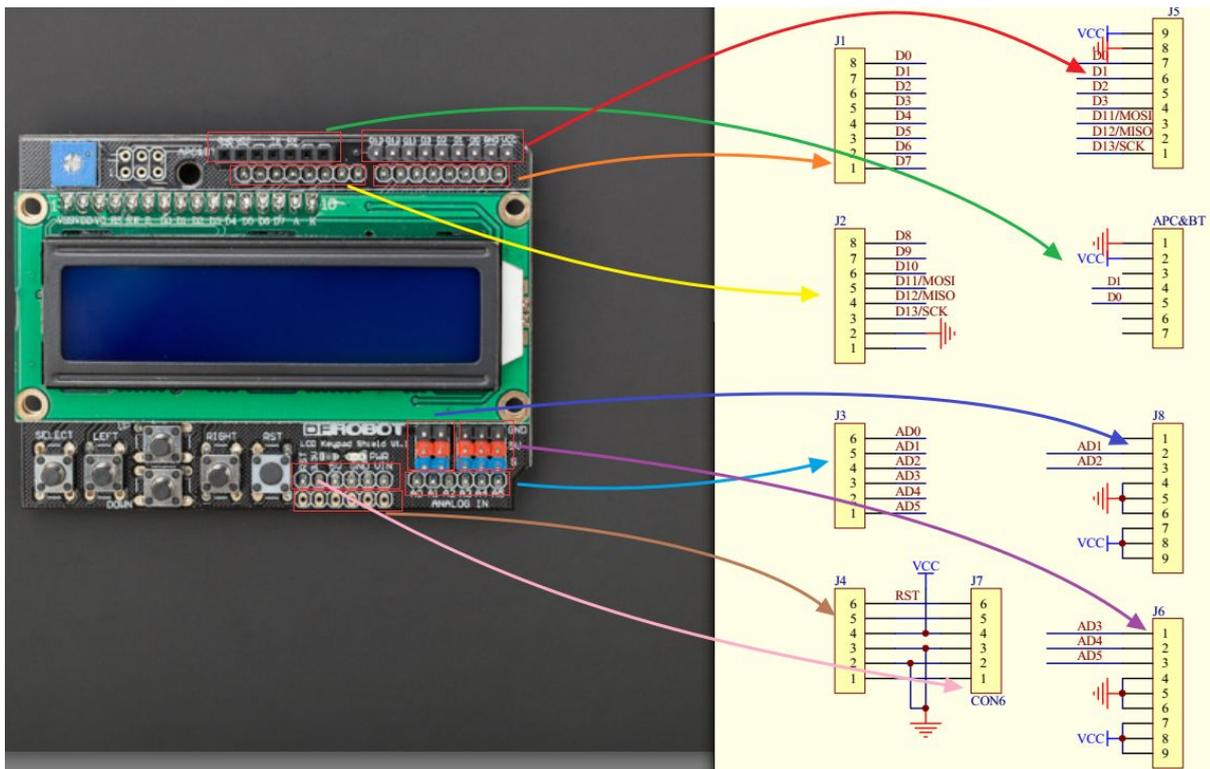
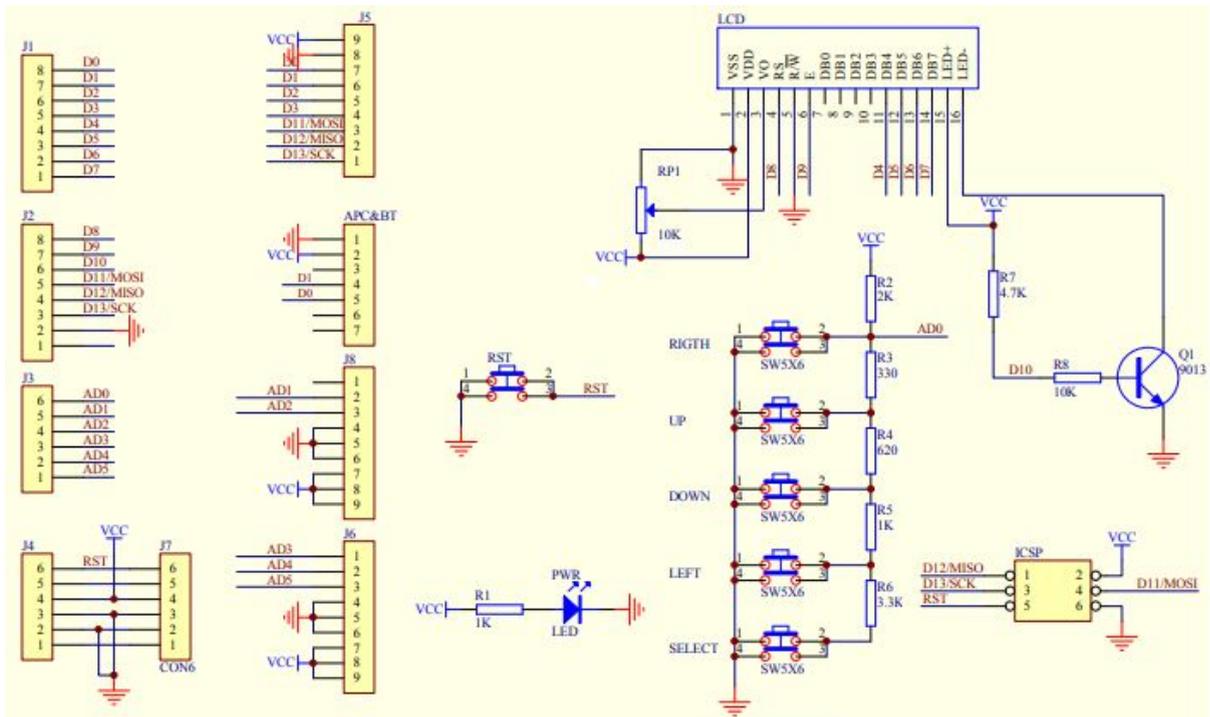
Fonte: ATMEL, 2014.

Anexo 02: Tabela com os pinos do atmega2560 e suas respectivas funções.

Pin Number	Pin Name	Mapped Pin Name	Pin Number	Pin Name	Mapped Pin Name
1	PG5 ( OC0B )	Digital pin 4 (PWM)	51	PG0 ( WR )	Digital pin 41
2	PE0 ( RXD0/PCINT8 )	Digital pin 0 (RX0)	52	PG1 ( RD )	Digital pin 40
3	PE1 ( TXD0 )	Digital pin 1 (TX0)	53	PC0 ( A8 )	Digital pin 37
4	PE2 ( XCK0/AIN0 )		54	PC1 ( A9 )	Digital pin 36
5	PE3 ( OC3A/AIN1 )	Digital pin 5 (PWM)	55	PC2 ( A10 )	Digital pin 35
6	PE4 ( OC3B/INT4 )	Digital pin 2 (PWM)	56	PC3 ( A11 )	Digital pin 34
7	PE5 ( OC3C/INT5 )	Digital pin 3 (PWM)	57	PC4 ( A12 )	Digital pin 33
8	PE6 ( T3/INT6 )		58	PC5 ( A13 )	Digital pin 32
9	PE7 ( CLKO/ICP3/INT7 )		59	PC6 ( A14 )	Digital pin 31
10	VCC	VCC	60	PC7 ( A15 )	Digital pin 30
11	GND	GND	61	VCC	VCC
12	PH0 ( RXD2 )	Digital pin 17 (RX2)	62	GND	GND
13	PH1 ( TXD2 )	Digital pin 16 (TX2)	63	PJ0 ( RXD3/PCINT9 )	Digital pin 15 (RX3)
14	PH2 ( XCK2 )		64	PJ1 ( TXD3/PCINT10 )	Digital pin 14 (TX3)
15	PH3 ( OC4A )	Digital pin 6 (PWM)	65	PJ2 ( XCK3/PCINT11 )	
16	PH4 ( OC4B )	Digital pin 7 (PWM)	66	PJ3 ( PCINT12 )	
17	PH5 ( OC4C )	Digital pin 8 (PWM)	67	PJ4 ( PCINT13 )	
18	PH6 ( OC2B )	Digital pin 9 (PWM)	68	PJ5 ( PCINT14 )	
19	PB0 ( SS/PCINT0 )	Digital pin 53 (SS)	69	PJ6 ( PCINT 15 )	
20	PB1 ( SCK/PCINT1 )	Digital pin 52 (SCK)	70	PG2 ( ALE )	Digital pin 39
21	PB2 ( MOSI/PCINT2 )	Digital pin 51 (MOSI)	71	PA7 ( AD7 )	Digital pin 29
22	PB3 ( MISO/PCINT3 )	Digital pin 50 (MISO)	72	PA6 ( AD6 )	Digital pin 28
23	PB4 ( OC2A/PCINT4 )	Digital pin 10 (PWM)	73	PA5 ( AD5 )	Digital pin 27
24	PB5 ( OC1A/PCINT5 )	Digital pin 11 (PWM)	74	PA4 ( AD4 )	Digital pin 26
25	PB6 ( OC1B/PCINT6 )	Digital pin 12 (PWM)	75	PA3 ( AD3 )	Digital pin 25
26	PB7 ( OC0A/OC1C/PCINT7 )	Digital pin 13 (PWM)	76	PA2 ( AD2 )	Digital pin 24
27	PH7 ( T4 )		77	PA1 ( AD1 )	Digital pin 23
28	PG3 ( TOSC2 )		78	PA0 ( AD0 )	Digital pin 22
29	PG4 ( TOSC1 )		79	PJ7	
30	RESET	RESET	80	VCC	VCC
31	VCC	VCC	81	GND	GND
32	GND	GND	82	PK7 ( ADC15/PCINT23 )	Analog pin 15
33	XTAL2	XTAL2	83	PK6 ( ADC14/PCINT22 )	Analog pin 14
34	XTAL1	XTAL1	84	PK5 ( ADC13/PCINT21 )	Analog pin 13
35	PL0 ( ICP4 )	Digital pin 49	85	PK4 ( ADC12/PCINT20 )	Analog pin 12
36	PL1 ( ICP5 )	Digital pin 48	86	PK3 ( ADC11/PCINT19 )	Analog pin 11
37	PL2 ( T5 )	Digital pin 47	87	PK2 ( ADC10/PCINT18 )	Analog pin 10
38	PL3 ( OC5A )	Digital pin 46 (PWM)	88	PK1 ( ADC9/PCINT17 )	Analog pin 9
39	PL4 ( OC5B )	Digital pin 45 (PWM)	89	PK0 ( ADC8/PCINT16 )	Analog pin 8
40	PL5 ( OC5C )	Digital pin 44 (PWM)	90	PF7 ( ADC7/TDI )	Analog pin 7
41	PL6	Digital pin 43	91	PF6 ( ADC6/TDO )	Analog pin 6
42	PL7	Digital pin 42	92	PF5 ( ADC5/TMS )	Analog pin 5
43	PD0 ( SCL/INT0 )	Digital pin 21 (SCL)	93	PF4 ( ADC4/TCK )	Analog pin 4
44	PD1 ( SDA/INT1 )	Digital pin 20 (SDA)	94	PF3 ( ADC3 )	Analog pin 3
45	PD2 ( RXDI/INT2 )	Digital pin 19 (RX1)	95	PF2 ( ADC2 )	Analog pin 2
46	PD3 ( TXD1/INT3 )	Digital pin 18 (TX1)	96	PF1 ( ADC1 )	Analog pin 1
47	PD4 ( ICP1 )		97	PF0 ( ADC0 )	Analog pin 0
48	PD5 ( XCK1 )		98	AREF	Analog Reference
49	PD6 ( T1 )		99	GND	GND
50	PD7 ( T0 )	Digital pin 38	100	AVCC	VCC

Fonte: adaptado ATMEL, 2014.

Anexo 03: Esquemático de ligação da LCD Keypad Shield.



Fonte: DFROBOT, sem ano.