



Aplicações do protocolo I²C em sistemas microcontrolados

Ítallo Santos Lima e Silva¹, Jorge Ricardo de Araujo Kaschny²

¹Graduando do Curso de Engenharia Elétrica – IFBA, Campus Vitória da Conquista. e-mail: itallosilva@gmail.com

²Professor Adjunto – IFBA, Campus Vitória da Conquista. e-mail: jrkaschny@physika.info

Resumo: Na presente contribuição apresentamos um conjunto de montagens didáticas relacionadas com a aplicação do protocolo I²C em sistemas microcontrolados. Para isso, empregamos microcontroladores AVR, fabricados pela ATMEL, e exploramos diversos dispositivos I²C facilmente encontrados no mercado. Em cada caso, foi possível estudar os princípios básicos de operação de tais dispositivos bem como efetuar a construção de pequenos protótipos que permitiram explorar a capacidade de cada um deles. Para o desenvolvimento do firmware utilizamos a plataforma de desenvolvimento BASCOM. Tais montagens podem ser facilmente aplicadas em atividades práticas nas disciplinas de eletrônica digital e microcontroladores, por um custo altamente convidativo.

Palavras-chave: microcontroladores, eletrônica digital, protocolo I²C.

1. INTRODUÇÃO

Atualmente, a utilização de microcontroladores se faz cada vez mais presente nas mais inusitadas áreas. Eles são utilizados em quase todos os equipamentos eletro-eletrônicos presentes em nosso cotidiano, possuindo as mais surpreendentes aplicações. Essa tendência tecnológica se deve aos grandes avanços da microeletrônica e dos diversos outros setores envolvidos no projeto e produção de dispositivos semicondutores. De fato, nos dias de hoje, o baixo custo, a facilidade na aquisição de componentes no mercado eletrônico e a existência de plataformas bastante eficientes para o desenvolvimento de programas (firmware), atuam como fatores motivadores para o desenvolvimento de novas aplicações. Tais fatores popularizaram algo que era basicamente restrito ao ambiente empresarial e acadêmico, viabilizando a criação de soluções em pequena escala por projetistas independentes. Assim, tendo em mente a flexibilidade e a gama de aplicações dos microcontroladores, é possível observar que em muitos casos o foco principal no desenvolvimento de um projeto passou gradualmente da construção de um circuito dedicado para a elaboração de um firmware conveniente.

Em termos práticos, o desenvolvimento de um projeto utilizando um microcontrolador comumente conduz o projetista a conviver e se acostumar com as limitações imposta por cada dispositivo. Durante o desenvolvimento de uma aplicação é comum nos depararmos com os problemas que envolvem a pequena quantidade de memória e principalmente o número finito de portas de entrada e saída (I/O). Portanto, a busca de soluções que procurem adequar tais limitações aos objetivos de um projeto se torna algo fundamental. Entre tais soluções encontramos os barramentos de comunicação serial, que são altamente convenientes visto que necessitam de um menor número de portas de I/O quando comparadas com barramentos paralelos.

Entre as diversas opções, temos o protocolo de comunicação serial I²C, que foi criado na década de 1990 pela Philips Semiconductor Division – atual NXP (NXP, 2012), tendo como propósito inicial a interconexão de microprocessadores a dispositivos controladores utilizados em equipamentos de áudio e principalmente televisores. A motivação da Philips partiu da necessidade imperativa de reduzir o número de vias de interconexão entre a central de processamento (cpu – microprocessador ou microcontrolador) e os demais dispositivos periféricos, com o objetivo de reduzir a complexidade dos circuitos e otimizar sua produção. Os estudos levaram então a criação de um barramento de 2 fios, denominado I²C (Inter- Integrated Circuit), ou seja, barramento para interconexão de circuitos integrados, que constitui uma interface simples, prática e padronizada.

Hoje em dia, temos a nossa disposição no mercado a mais variada gama de dispositivos periféricos utilizando o protocolo I²C. Tal variedade abrange extensores de portas de I/O, conversores analógico-digital (ADC), conversores digital-analógico (DAC), módulos relógio/calendário, sensores,

teclados, módulos TCP/IP, mostradores (displays), controladores de áudio analógico e até mesmo sintonizadores completos de rádio e televisão. Dessa maneira, tendo em mente tais considerações, apresentamos a seguir um conjunto de montagens didáticas relacionadas com aplicações do protocolo I²C em sistemas microcontrolados, procurando explorar e divulgar as suas potencialidades.

2. MATERIAIS E MÉTODOS

O barramento I²C consiste fisicamente de duas vias. A primeira delas corresponde ao relógio (clock) de comunicação – chamada SCL (serial clock line), sendo a segunda responsável pela comunicação bidirecional de dados - SDA (serial data line). Nesse tipo de protocolo, o dispositivo que inicia a comunicação é chamado de mestre (master), sendo os demais, que são por ele controlados, chamados de escravos (slaves). O clock de comunicação é comumente ditado pelo dispositivo mestre, sendo a troca bidirecional de informações evidentemente efetuada de forma serial. Tais informações podem ser constituídas de dados, endereços ou mesmo comandos. Dessa forma, o dispositivo mestre pode efetuar tanto uma escrita como uma leitura via a linha SDA. Uma visão geral desse barramento é ilustrada na figura 1, onde GND e VCC representam as linhas de alimentação (0/5V) dos dispositivos.

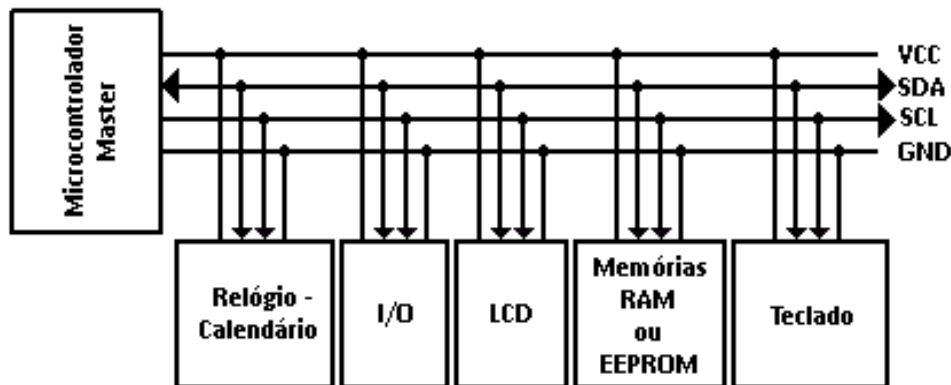


Figura 1 – Diagrama ilustrando um barramento I²C. A linha SCL corresponde à via de clock e a SDA a linha bidirecional de dados. GND e VCC correspondem à alimentação (0/5V).

A comunicação com os dispositivos escravos é sempre inicializada pelo dispositivo mestre. A identificação é feita pelo endereço de cada dispositivo escravo. Tal endereço é em parte fixado pelo fabricante, possibilitando ao projetista somente um certo ajuste. O protocolo utiliza-se das duas linhas para efetuar a comunicação, ou seja, os critérios de partida e de parada do fluxo, assim como os dados a serem enviados ou recebidos pelo mestre, são controlados pelas linhas SCL e SDA. O critério de partida e parada é determinado pela variação do estado da linha de dados, estando a linha de clock em nível alto. Para o envio de dados, é necessário que haja uma mudança de estado da linha SDA no momento em que a linha de clock estiver em nível baixo, sendo o dado validado no ponto de transição da linha de clock (de baixo para alto). Resumidamente, o mestre inicia a comunicação com a condição de partida, envia ao barramento o endereço do dispositivo que deseja se comunicar, seleciona entre leitura ou escrita, aguarda validação e a partir deste momento troca dados com o dispositivo escravo até que seja enviada a condição de parada. Tal fluxo é ilustrado esquematicamente abaixo.

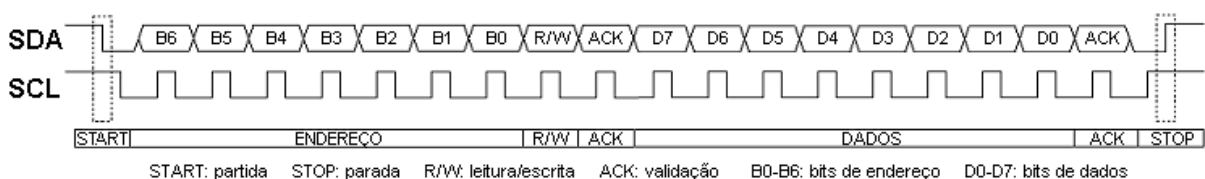


Figura 3 – Diagrama geral ilustrando o fluxo de endereço e dados em um barramento I²C.



Como mencionado anteriormente, devido a sua simplicidade, o protocolo I²C foi largamente aceito pelo mercado e adotado por diversos fabricantes de semicondutores tais como Atmel, Intel, Texas, etc, além da própria NXP. Uma das vantagens de se utilizar o barramento em aplicações microcontroladas está na economia de portas de I/O do microcontrolador, pois para o seu funcionamento são necessários somente dois pinos. Isso contribui significativamente para simplificar o projeto de um circuito microcontrolado, dispensando a utilização de estratégias mais complicadas para multiplexação das portas de I/O. Além disso, o I²C possibilita a conexão de um número significativo de dispositivos em um mesmo barramento (em princípio até 128) sem a necessidade da inclusão de um hardware adicional. Tendo em mente que o clock é ditado pelo mestre, uma outra vantagem está no fato de que a comunicação não fica limitada a uma taxa de transmissão (baud rate) fixa sendo, portanto, algo flexível. Porém, é típico que a comunicação tenha a tendência de ser mais lenta. Na versão padrão do protocolo I²C, que é a mais encontrada, a velocidade de comunicação é prevista para atingir até 100 kbits/s. Portanto, é comum falarmos que o I²C é realmente eficiente somente para aquelas aplicações onde a velocidade do fluxo de informações não é um fator crítico ou determinante. Contudo, cabe mencionar que revisões atualizadas citam a implementação de um modo de alta velocidade que pode atingir 3.4 Mbits/s. Entretanto, dispositivos com tais características são atualmente bem menos frequentes no mercado comum.

Nas montagens e testes que realizamos foi utilizado como mestre o microcontrolador ATmega8, fabricado pela Atmel (ATMEL, 2012), com o qual já possuímos uma razoável experiência e boa parte da infra-estrutura necessária. Obviamente o uso do protocolo I²C não está limitado a essa família de microcontroladores, podendo ser usado em conjunto com qualquer microcontrolador disponível no mercado. A elaboração do firmware foi efetuada utilizando a linguagem basic, via a plataforma de desenvolvimento BASCOM (MCS, 2012), conjuntamente com um gravador AVR-910/USB (PHYSIKA, 2012), desenvolvido previamente. A escolha de tal plataforma se deu originalmente pela variedade de recursos oferecidos, onde são combinados uma linguagem de rápido aprendizado com um substancial número de bibliotecas dedicadas às mais variadas finalidades. Evidentemente, outras plataformas encontram-se disponíveis na Internet, como por exemplo, o ATMEL STUDIO (ATMEL, 2012) que opera em conjunto com o compilador de código aberto AVR-GCC (WINAVR, 2012).

Os dispositivos I²C utilizados foram: (i) um extensor de entrada e saída de 8 bits - PCF8574 (PHILIPS, 2002), (ii) memórias EEPROM com 512 kBits - AT24C512B (ATMEL, 2012) e (iii) um relógio de tempo real com calendário - PCF8583 (NXP, 2010). Adicionalmente, exploramos o desenvolvimento de um display operando como periférico I²C. Para isso empregamos um PCF8574 em conjunto com um display LCD alfanumérico com 16 colunas e 4 linhas - JHD539-164B (SHENZHEN, 2012). Para os testes de operação utilizamos a porta serial disponível no microcontrolador para se comunicar com um computador pessoal (PC). De fato, a alimentação do protótipo foi fornecida pela porta USB de tal PC. Toda a comunicação entre o protótipo microcontrolado e o microcomputador foi efetuada usando um conversor UART/USB – FT232R (FTDI, 2012), em conjunto com o HyperTerminal (ou similar) disponível no Windows. A escolha de tal conversor se deve não somente a uma questão de conveniência, com também a disponibilidade de drivers, oferecidos pelo fabricante para os sistemas operacionais Windows e Linux.

3. RESULTADOS E DISCUSSÃO

Como visto, cada dispositivo escravo, conectado ao barramento, deve possuir um endereço específico. Esse endereço consiste em uma seqüência de oito bits, onde tipicamente os quatro bits mais significativos são configurados pelo fabricante. Os três bits seguintes são configurados pelo usuário, via pinos específicos, disponíveis no circuito integrado. O bit menos significativo é reservado para selecionar entre operações de leitura e escrita no dispositivo escravo. Com a variação dos bits disponíveis ao usuário é possível obter até oito combinações diferentes de endereço, possibilitando a incorporação de até oito dispositivos idênticos em um mesmo barramento. Contudo, são observadas algumas variações que dependem especificamente do dispositivo. Este é o caso do PCF8583 que disponibiliza apenas um pino para configuração do endereço. Para melhor ilustrar esses detalhes, apresentamos na tabela a seguir as características do endereçamento dos dispositivos utilizados.

Tabela 1 – Endereçamento dos dispositivos I²C utilizados.

Dispositivo	Especificação do Endereço			Faixa de Endereçamento
	Fabricante	Usuário *	Tipo de Operação **	
PCF8574	0 1 0 0	A2 A1 A0	0	de 01000000 a 01001110
	0 1 0 0	A2 A1 A0	1	de 01000001 a 01001111
AT24C512B	1 0 1 0	A2 A1 A0	0	de 10100000 a 10101110
	1 0 1 0	A2 A1 A0	1	de 10100001 a 10101111
PCF8583***	1 0 1 0 0 0 0	A0	0	10100000 ou 10100010
	1 0 1 0 0 0 0	A0	1	10100001 ou 10100011

* O usuário pode selecionar o estado de cada pino (A0, A1 ou A2) de acordo com a necessidade do projeto.

** Este constitui o bit menos significativo e seleciona entre as operações de escrita (0) ou leitura (1).

*** Em particular, este dispositivo possui somente um pino (A0) para configuração do endereço.

Uma visão geral do protótipo pode ser vista no diagrama esquemático mostrado na figura 2. Como podemos observar, empregamos aqui um microcontrolador ATmega8 como dispositivo mestre e incluímos todos os dispositivos acima mencionados de forma a ilustrar a aplicabilidade do protocolo I²C no controle de diversos dispositivos.

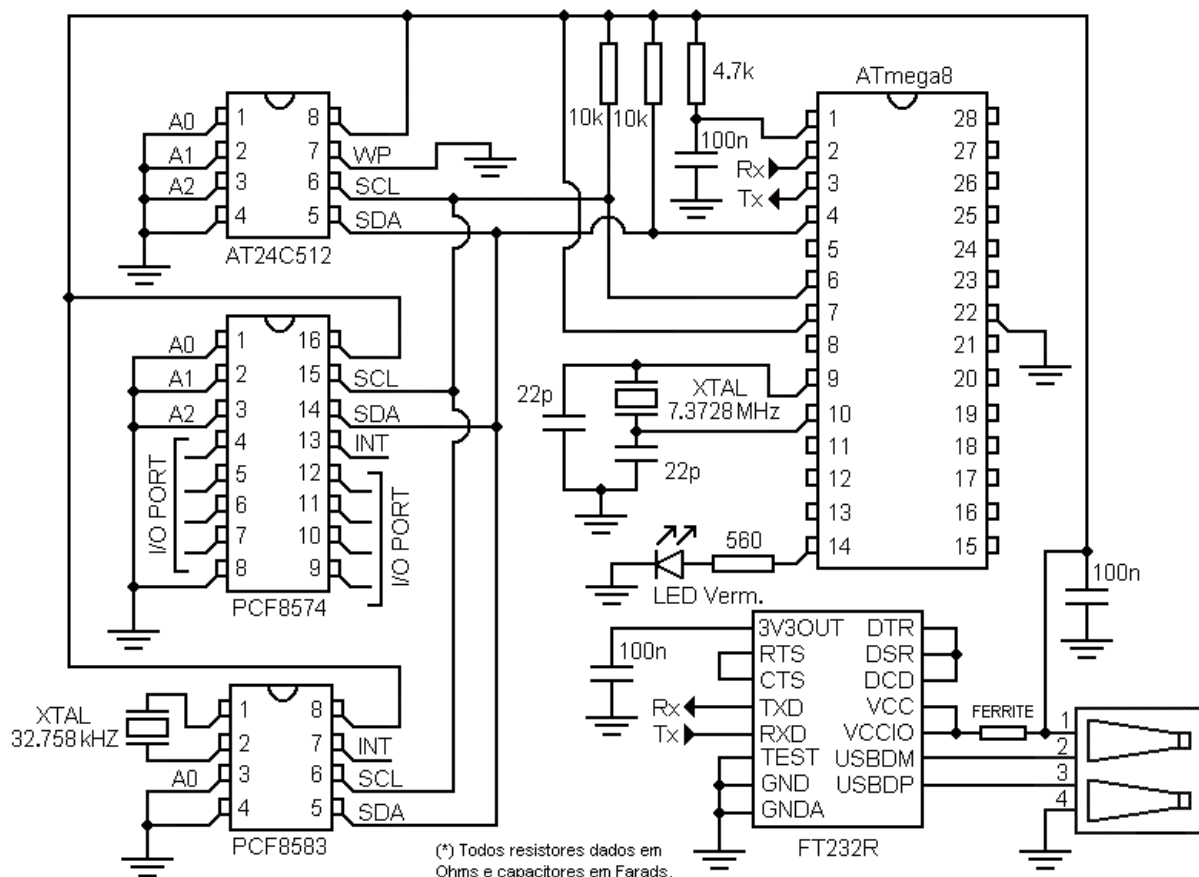


Figura 3 – Diagrama esquemático do sistema microcontrolado usando barramento I²C.

Para a elaboração do firmware, utilizando a plataforma de desenvolvimento BASCOM, são usadas as rotinas da biblioteca i2c.lib (ou i2c.lbx), incluídas no pacote do aplicativo. Salienta-se que as particularidades na passagem e no fluxo de instruções, bem como na leitura e escrita de dados, devem

observar as especificações técnicas de cada dispositivo escravo. Contudo, em termos gerais, e após a configuração dos pinos SCL e SDA, a comunicação segue os seguintes passos: (i) O mestre inicializa o barramento utilizando o comando I2cstart; (ii) Envia um byte contendo o endereço do dispositivo que deseja se comunicar via o comando I2cwbyte, especificando também o tipo de operação que será a seguir efetuada, ou seja, leitura ou escrita; (iii) Após a confirmação do recebimento do comando por parte dispositivo escravo, algo que é transparente ao programador, o mestre executa o comando de escrita (I2cwbyte) ou leitura (I2crbyte) correspondentemente com o especificado na etapa ii, e; (iv) Finaliza o barramento via o comando I2cstop. Nos casos onde é necessário alternar operações de leitura e escrita devemos observar que o barramento deve ser reinicializado convenientemente. Um exemplo de tal situação é o caso onde desejamos efetuar uma leitura do conteúdo de um determinado endereço armazenado na memória EEPROM. Na primeira etapa o barramento é inicializado para escrita, escrevendo-se em seguida os bits mais significativos do endereço a ser lido seguido pela escrita dos bits menos significativos. Após tal etapa, reinicializa-se o barramento para leitura, sem que o mesmo tenha sido finalizado. Assim, efetua-se a respectiva operação leitura do endereço desejado e, somente após tal passo, finaliza-se o barramento.

Como sugestões de aplicações imediatas do barramento I²C, apresentamos uma expansão de memória não volátil, com capacidade de 8x512 kbits, e um display alfanumérico operado via tal barramento. Na figura 4 temos o diagrama esquemático da expansão de memória, onde devemos observar a alternância entre a ligação dos pinos A0, A1 e A2 ao GND e VCC, em cada chip de memória, de maneira a compor um banco cujo endereçamento ocupa toda a faixa indicada na tabela 1.

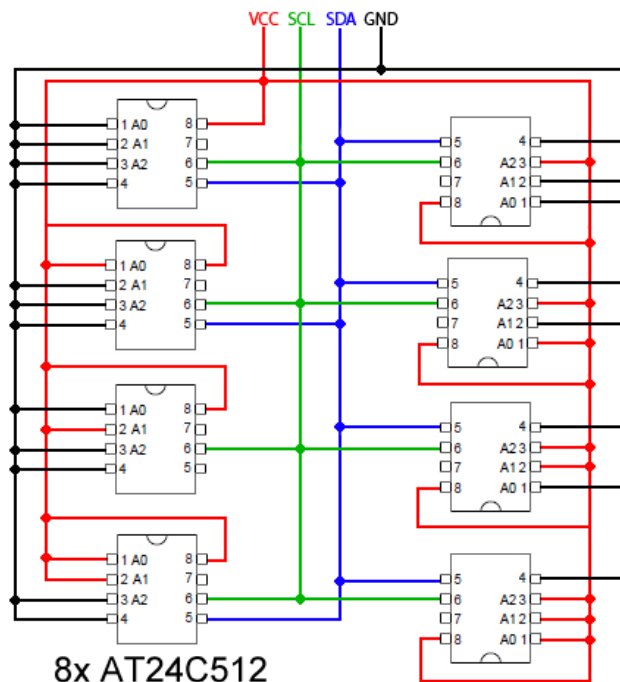


Figura 4 – Diagrama esquemático de um banco de memória não volátil usando barramento I²C.

Correspondentemente, na figura 5, é mostrado o diagrama esquemático de um display alfanumérico I²C composto pela combinação do módulo JHD539-164B e do extensor de I/O PCF8574. Salienta-se que neste caso o controle de tal display passa a ser feito no BASCOM via as rotinas disponibilizadas na biblioteca lcd_i2c.lib (ou lcd_i2c.lbx), sem maiores repercussões na tarefa de programação. Tal aplicação oferece um comparativo bastante conveniente das vantagens do barramento I²C. Tipicamente um display alfanumérico emprega no mínimo 6 portas de I/O, ou seja, usualmente ele necessita 4 portas a mais que o necessário para implementação do mesmo recurso utilizando a abordagem esquematizada na figura 5. Portanto, fica claramente exemplificado que a aplicação do

barramento I²C é refletida diretamente na “economia” de portas de I/O, algo bem importante a ser considerado durante o projeto de um sistema microcontrolado. Adicionalmente, cabe observar que o display I²C esquematizado na figura abaixo pode ser diretamente conectado ao circuito da figura 3, constituindo um bom exemplo da flexibilidade do barramento.

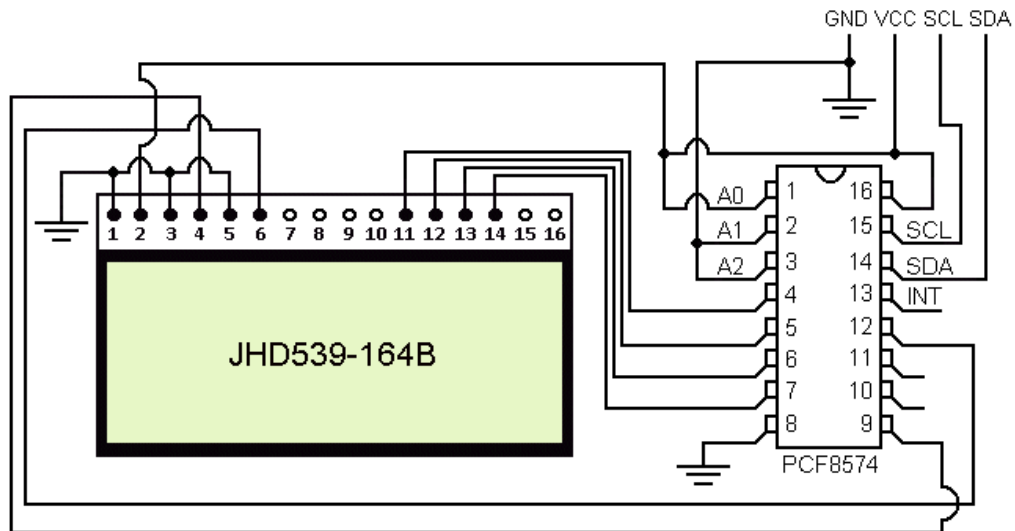


Figura 5 – Diagrama esquemático de um display controlado via protocolo I²C.

Para ilustrar outra aplicação prática, mostramos na figura 6 a combinação do display acima descrito com um PCF8583 em uma montagem experimental de um relógio/calendário microcontrolado. O PCF8583 pode ser eventualmente substituído pelo DS1307 que também é um relógio/calendário I²C. Salienta-se que tais dispositivos constituem uma das maneiras mais práticas de se obter as funcionalidades típicas de um relógio e de um calendário, sendo sua presença bem usual em diversos eletrodomésticos.

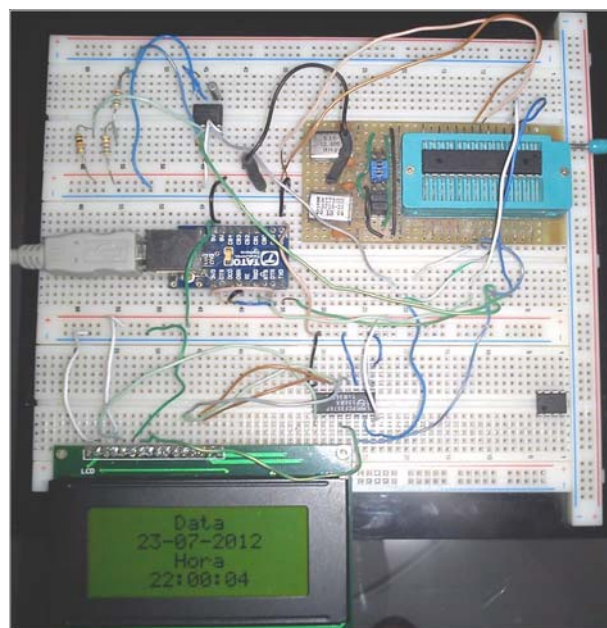


Figura 6 – Montagem experimental de um relógio/calendário utilizando o protocolo I²C.



Finalizando, ressaltamos a importância da inclusão de atividades práticas em disciplinas que abordam o desenvolvimento de sistemas microcontrolados. Sendo assim, atividades que visem a divulgação das diversas tecnologias e recursos disponíveis, entre eles o protocolo I²C, são de grande utilidade. Nesse contexto, as montagens apresentadas e as informações aqui divulgadas podem contribuir positivamente para tal esforço. Tendo em mente que todas as montagens comentadas acima foram efetuadas usando um protoboard, o custo de implementação em um laboratório didático consistirá basicamente do custo dos componentes empregados e da construção do gravador. Apesar da plataforma BASCOM ser um software comercial, a versão disponível para download permite o desenvolvimento de firmwares com até 4 kbytes, o que preenche muito bem a maioria das necessidades usuais. Alternativamente é possível empregar a plataforma gratuita ATMEL STUDIO, que utiliza o compilador de código livre AVR-GCC, fazendo uso da biblioteca twi.h para comunicação via I²C. Indo além, é possível o projeto futuro de um kit de desenvolvimento que disponibilize, entre outros recursos, tal protocolo.

4. CONCLUSÕES

Em resumo, apresentamos na presente contribuição um conjunto de montagens didáticas relacionadas com aplicações do protocolo I²C em sistemas microcontrolados. Em cada caso, foi possível estudar os princípios básicos de operação de tais dispositivos, bem como efetuar a construção de pequenos protótipos que permitiram explorar a capacidade de cada um deles. Tais montagens podem ser facilmente aplicadas em atividades práticas nas disciplinas de eletrônica digital e microcontroladores, por um custo altamente convidativo.

AGRADECIMENTOS

Agradecemos ao IFBA, Instituto Federal da Bahia, e a FAPESB, Fundação de Amparo a Pesquisa do Estado da Bahia, pelo apoio ao presente trabalho via a concessão de uma bolsa através do Programa Institucional de Bolsas de Iniciação Científica (PIBIC).

REFERÊNCIAS

ATMEL. **AVR Microcontrollers**. Disponível em: <<http://www.atmel.com>> Acesso em 8 ago 2012.

ATMEL. **Serial EEPROM**. Disponível em: <<http://www.atmel.com>> Acesso em 8 ago 2012.

FTDI. **FT232R USB UART I.C.** Disponível em: <<http://www.ftdichip.com>> Acesso em 8 ago 2012.

MCS. **Bascom AVR**. Disponível em: <<http://www.mcselec.com>> Acesso em 8 ago 2012.

NXP. **PCF8582 - Data Sheet**. Disponível em: <<http://www.nxp.com>> Acesso em 8 ago 2012.

NXP. **I²C Specification and User Manual**. Disponível em:
<http://www.nxp.com/documents/user_manual/UM10204.pdf> Acesso em 8 ago 2012.

PHILIPS. **PCF8574 - Data Sheet**. Disponível em: <<http://www.nxp.com>> Acesso em 8 ago 2012.

PHYSIKA. **Programador AVR910 Versão USB**. Disponível em:
<<http://physika.info/physika/index.php/progavrusb.html>> Acesso em 8 ago 2012.

SHENZHEN. **JHD539-164B - 16X4 Characters - Data Sheet**. Disponível em
<<http://www.alldatasheet.com/datasheet-pdf/pdf/276143/JHD/JHD539-164B.html>> Acesso em 8 ago 2012.

WINAVR, **Open Source Software Development Tools for the Atmel AVR Microcontrollers**. Disponível em <<http://winavr.sourceforge.net>> Acesso em 8 ago 2012.