

**Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP)
Faculdade de Ciências
Departamento de Computação
Campus de Bauru – SP**

**TRABALHO DE CONCLUSÃO DA DISCIPLINA
MICROCONTROLADORES (TCD)**

Disciplina: Microcontroladores

Aluna: Heloisa Hungaro Primolan RA: 141026431

E-mail: heloisa.hungaro@gmail.com

Professor: João E. M. Perea Martins, Prof. Dr.

Data: 02 jul. 2016

Trabalho apresentado como parte das atividades da disciplina Microcontroladores do curso de graduação de Bacharelado em Ciência da Computação, Faculdade de Ciências - UNESP.

Bauru, 2016

1 OBJETIVO E DESCRIÇÃO DO PROJETO

O objetivo deste projeto é o desenvolvimento de um sistema automatizado que controla a temporização de um semáforo em um cruzamento, levando em consideração a ocorrência de chuvas. Em algumas regiões existe o problema de enchentes e algumas vias são evitadas ou mesmo interditadas temporariamente quando ocorrem chuvas, o que causa um aumento significativo de fluxo de carros em outras ruas, gerando problemas de congestionamento. Assim, esse sistema realiza o controle dos tempos dos semáforos de um cruzamento, de modo que o semáforo da via utilizada como escape, ou seja, a via em que há maior movimento em períodos de chuva, permaneça aberto (verde) por um tempo maior em relação ao semáforo da via menos utilizada, de modo a favorecer o fluxo de veículos nos períodos de ocorrência de chuva.

2 MATERIAIS DO PROJETO

Para desenvolvimento do projeto, foram utilizados os materiais listados abaixo.

- a) 1 microcontrolador PIC16F873A (28 pinos)
- b) 6 *LEDs* vermelhos
- c) 2 *LEDs* amarelos
- d) 2 *LEDs* verdes
- e) 10 resistores de 330 Ω (um para cada *LED*)
- f) 1 sensor de chuva (com saída analógica e digital)
- g) 1 fonte de 5 *Volts*
- h) 1 placa auxiliar para desenvolvimento *PIC*
- i) 1 *protoboard*
- j) 1 gravador *USB* para *PIC*
- k) 1 cabo *USB – UART* (para comunicação serial)
- l) Diversos fios *jumpers* macho/macho

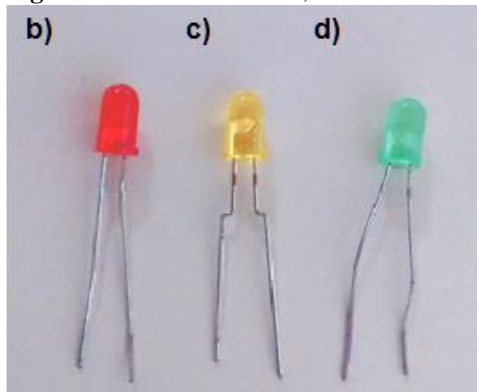
As figuras 1 a 9 mostram os componentes utilizados neste projeto, identificados conforme a lista acima (alfabeticamente).

Figura 1 – Microcontrolador PIC16F873A



Fonte: Página web da empresa *FLIPEFLOP*¹

Figura 2 – *LEDs* vermelho, amarelo e verde



Fonte: própria autora (2016)

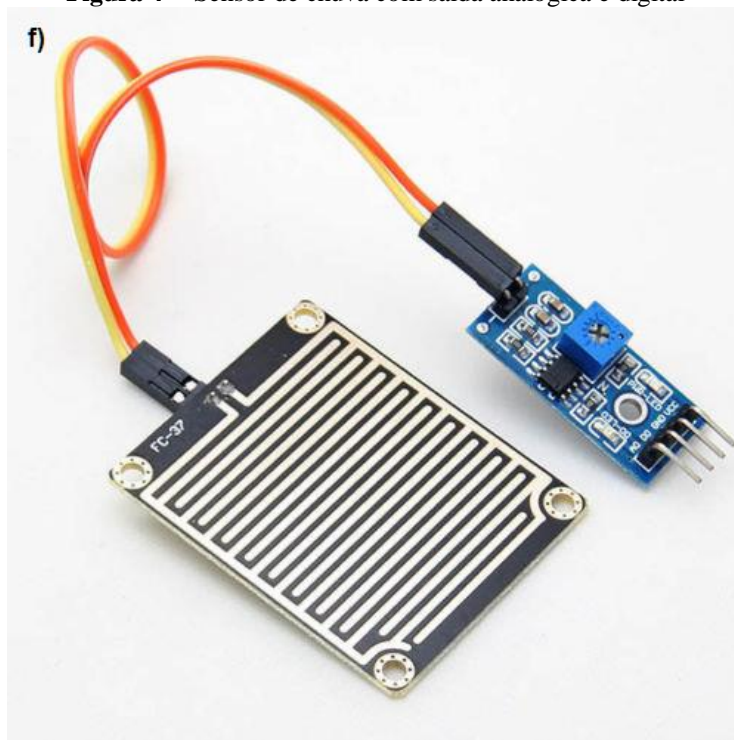
¹ Disponível em <www.flipeflop.com/pd-274f89-microcontrolador-pic-16f873a.html>. Acesso em 27 jun. 2016.

Figura 3 – Resistor de 330 Ω (laranja/laranja/marrom)



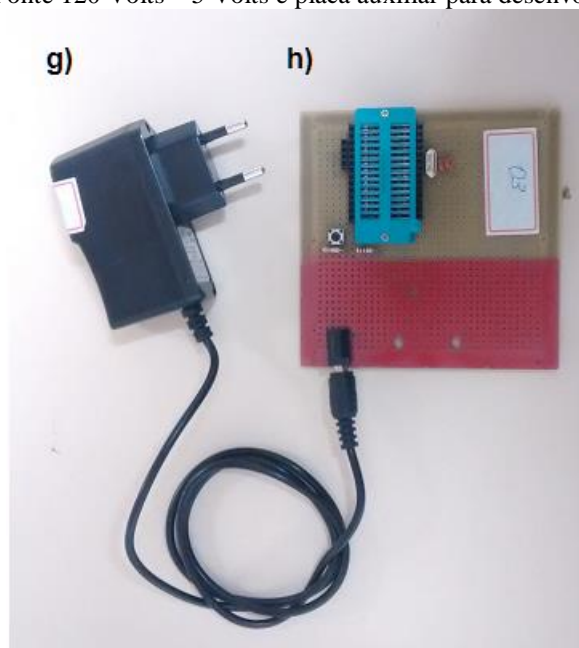
Fonte: própria autora (2016)

Figura 4 – Sensor de chuva com saída analógica e digital



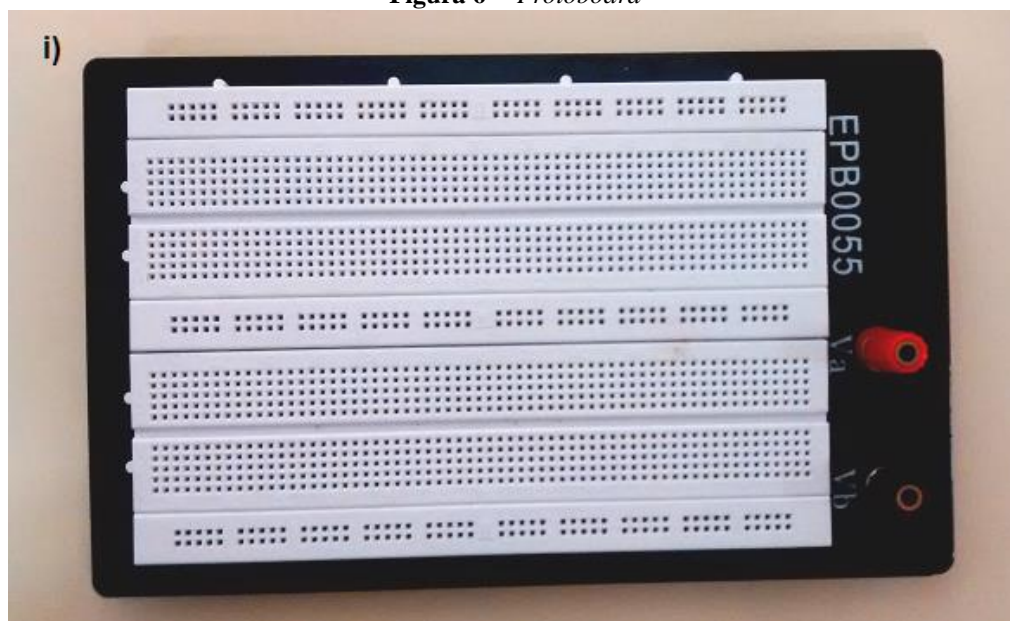
Fonte: Página web da empresa ACLINFO²

Figura 5 – Fonte 120 Volts – 5 Volts e placa auxiliar para desenvolvimento PIC



Fonte: própria autora (2016)

² Disponível em <www.aclinformatica.com/products.php?product=sensor-de-chuva-arduino>. Acesso em 27 jun. 2016.

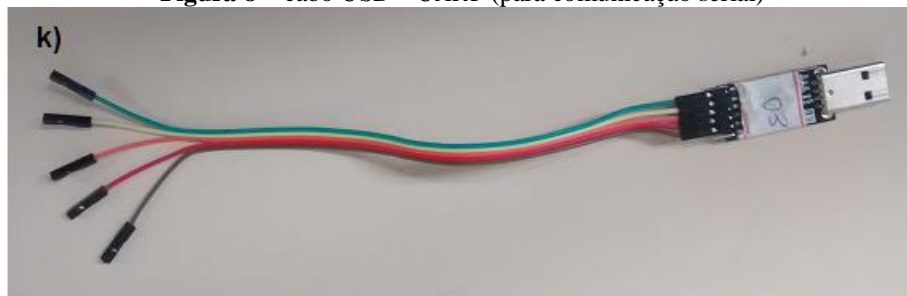
Figura 6 – Protoboard

Fonte: própria autora (2016)

Figura 7 – Gravador USB para PIC

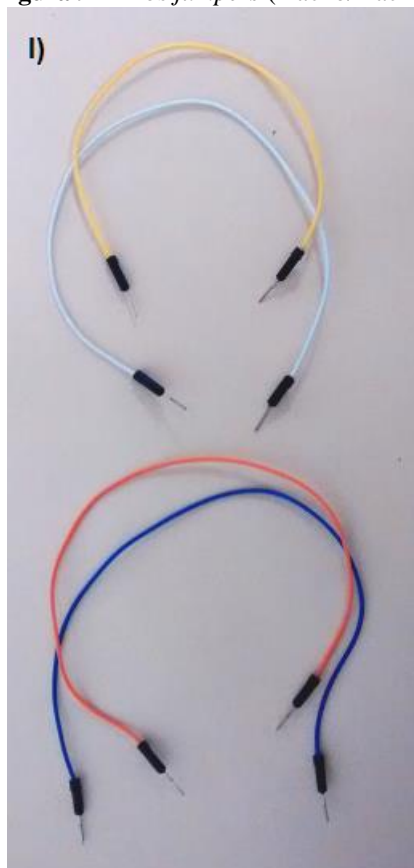
Fonte: própria autora (2016)

Figura 8 – cabo *USB – UART* (para comunicação serial)



Fonte: própria autora (2016)

Figura 9 – Fios *jumper*s (macho/macho)



Fonte: própria autora (2016)

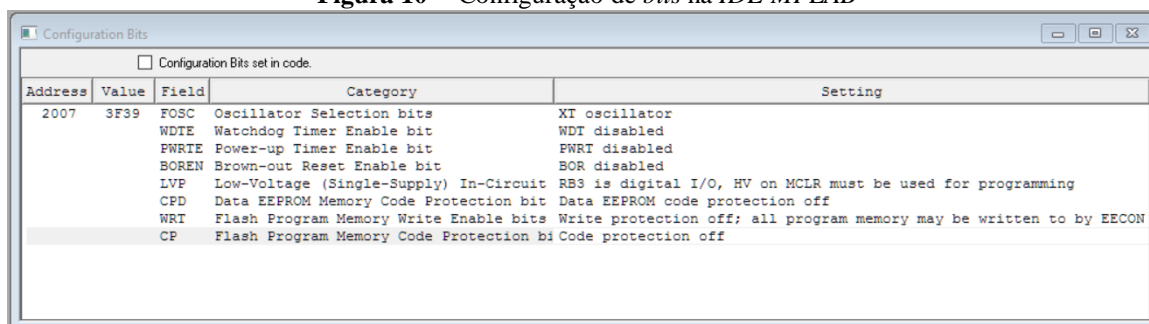
3 DETALHES OPERACIONAIS E FUNCIONAMENTO DO SISTEMA

Foram desenvolvidos dois códigos para o funcionamento do sistema: um para saída digital do sensor de chuva e outro para saída analógica. Os códigos apresentam grande semelhança, sendo o segundo mais complexo e de melhor aplicabilidade.

A IDE utilizada foi a *MPLAB* (versão 8.92), com programador no modo *PICKit2*, suportada pelo gravador utilizado (*MultiPROG Plus*).

A configuração de *bits* no *MPLAB* deve ser programada conforme apresentado na figura 10.

Figura 10 – Configuração de *bits* na IDE *MPLAB*



The screenshot shows the 'Configuration Bits' window in MPLAB. A checkbox at the top indicates 'Configuration Bits set in code.' is checked. Below is a table with columns: Address, Value, Field, Category, and Setting.

| Address | Value | Field | Category | Setting |
|---------|-------|-------|--|---|
| 2007 | 3F39 | FOSC | Oscillator Selection bits | XT oscillator |
| | | WDTE | Watchdog Timer Enable bit | WDT disabled |
| | | PWRT | Power-up Timer Enable bit | PWRT disabled |
| | | BOREN | Brown-out Reset Enable bit | BOR disabled |
| | | LVP | Low-Voltage (Single-Supply) In-Circuit | RB3 is digital I/O, HV on MCLR must be used for programming |
| | | CPD | Data EEPROM Memory Code Protection bit | Data EEPROM code protection off |
| | | WRT | Flash Program Memory Write Enable bits | Write protection off; all program memory may be written to by EECON |
| | | CP | Flash Program Memory Code Protection bit | Code protection off |

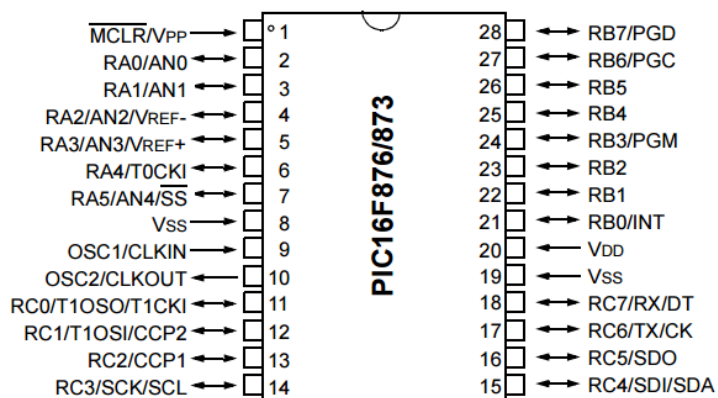
Fonte: própria autora (2016)

Como o projeto necessita de controle temporal, utilizamos o *timer* TMR0 do PIC com a seguinte configuração: 131 para o TMR0 e 32 para o *Prescaler*. Desta forma, obtemos uma frequência de 250 interrupções por segundo.

Para fins de entendimento, denominaremos os semáforos de semáforo 1 (s1) e semáforo 2 (s2), sendo o s1 (o semáforo da via utilizada como escape) o primeiro a abrir (ficar verde) no cliço de semáforos. Consideraremos um ciclo de semáforos como o período desde o s1 tornar-se verde, seguido de amarelo e vermelho, somado ao mesmo período para o s2. Assim, quando o semáforo 1 torna-se verde novamente, significa o início de um novo ciclo de semáforos.

3.1 PINAGENS: PIC, SENSOR DE CHUVA E SERIAL

As figuras 11, 12 e 13 mostram as pinagens do PIC16F873A, do cabo para comunicação serial e do sensor de chuva, respectivamente.

Figura 11 – Pinagem do PIC16F873A

Fonte: Data Sheet do PIC16F873A da MICROCHIP³

Figura 12 – Pinagem do cabo para comunicação serial

Fonte: própria autora (2016)

Figura 13 – Pinagem do sensor de chuva

Fonte: própria autora (2016)

³ Disponível em <ww1.microchip.com/downloads/en/DeviceDoc/30292D.pdf>. Acesso em 27 jun. 2016.

3.2 SISTEMA PARA SAÍDA DIGITAL

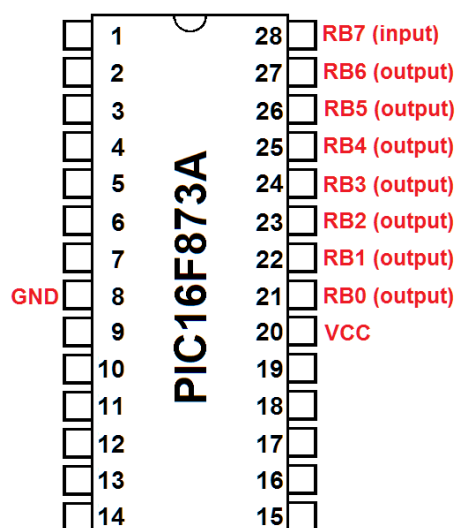
Utiliza a saída digital do sensor de chuva para obter a informação de que está ou não chovendo. Não requer conversão Analógico/Digital (A/D) nem comunicação serial.

O sensor de chuva envia sinal baixo (0) quando há chuva e sinal alto (1) quando não há chuva.

3.2.1 Pinagem

A figura 14 abaixo informa quais pinos do PIC foram utilizados no projeto digital.

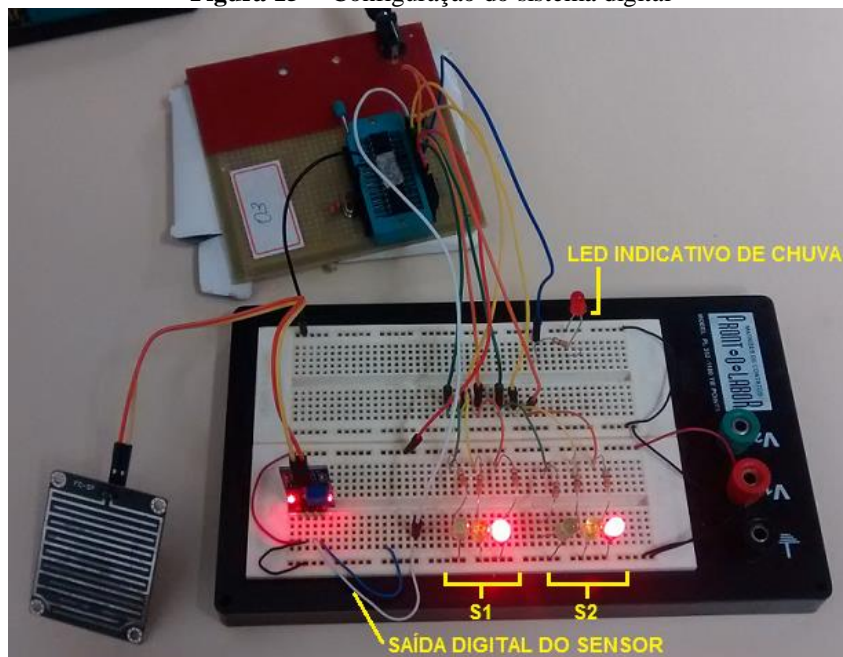
Figura 14 – Pinos utilizados no sistema digital



Fonte: própria autora (2016)

O pino GND foi conectado à fonte terra presente no *protoboard*; o pino VCC foi conectado ao *protoboard* para fornecer uma fonte de alimentação de 5 Volts para o sensor de chuva; o pino RB7 foi utilizado para receber o sinal digital (1 ou 0) do sensor de chuva; o pino RB0 foi utilizado para enviar dados ao *LED* indicador de chuva; já os pinos RB1 a RB6 foram utilizados para enviar dados aos *LEDs* verde (RB1), amarelo (RB2) e vermelho (RB3) do semáforo 1 e verde (RB4), amarelo (RB5) e vermelho (RB6) do semáforo 2.

O sistema montado ficará conforme a figura 15.

Figura 15 – Configuração do sistema digital

Fonte: própria autora (2016)

3.2.2 Funcionamento

O sistema digital apresenta dois ciclos padronizados: o ciclo sem chuva e o ciclo com chuva.

O ciclo sem chuva é temporizado tal que, do s1, o sinal verde fica aceso durante 10 segundos, o amarelo durante 4 segundos e o vermelho durante 2 segundos (período em que o s2 vermelho permanece aceso) e, para o s2, ocorre a mesma temporização posterior (período em que o s1 vermelho permanece aceso).

O ciclo com chuva difere-se do ciclo sem chuva quanto aos tempos dos sinais verdes. Neste ciclo, o sinal verde do s1 permanece aberto durante 15 segundos e o do s2 durante 5 segundos. Isso facilita o fluxo na via utilizada como escape (via que apresenta o s1).

Para controle sobre qual ciclo será efetuado, foi criada uma variável que conta se está chovendo há mais de 30 segundos ou não. Ela apresenta valor 0 quando está chovendo há mais de 30 segundos, qualquer valor entre 1 e 29 se está chovendo há menos de 30 segundos ou valor 30 se não estiver chovendo.

Mas como isso funciona? A variável é iniciada com valor 30. A cada segundo, o *PIC* verifica se o sinal recebido pela saída digital do sensor de chuva é alto ou baixo. Se o sinal for alto, a variável recebe valor 30, pois não está chovendo. Se o sinal for baixo, verifica-se se o valor da variável já é 0 e, caso não seja, seu valor é decrescido em 1. Além disso, a cada

segundo testa-se se o valor dessa variável é igual a 0 e, caso seja, o *LED* indicativo de chuva é aceso; caso não seja, é apagado. Isso permite o acompanhamento visual do funcionamento do sistema.

A cada final de ciclo, tanto do ciclo com chuva quanto do ciclo sem chuva, testa-se se essa variável apresenta valor 0. Caso presente, o próximo ciclo a ser executado será o ciclo com chuva; caso não presente, o próximo será o ciclo sem chuva.

Essa sequência de testes é realizada infinitamente, permitindo a atualização da temporização dos semáforos de acordo com a ocorrência ou não de chuva em determinado período.

É importante ressaltar que esse sistema não leva em consideração a intensidade da chuva, pois o sinal digital não nos permite acesso nenhum a essa informação, diferentemente do sinal analógico.

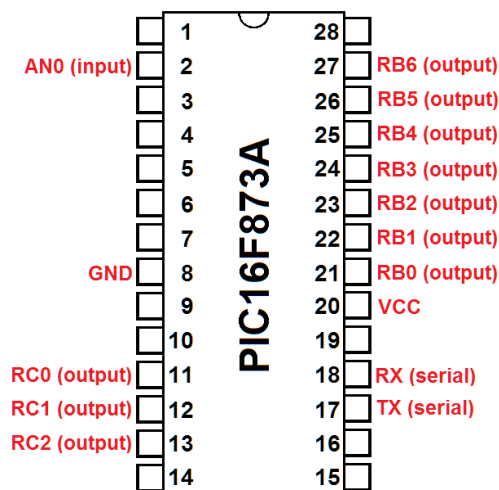
3.3 SISTEMA PARA SAÍDA ANALÓGICA

Utiliza a saída analógica do sensor de chuva para obter a informação de que está ou não chovendo e, caso esteja, poder definir se está chovendo razoavelmente ou bastante a definir pelo valor encontrado na conversão A/D. A comunicação serial é utilizada para visualização dos valores recebidos pela saída analógica do sensor de chuva.

3.3.1 Pinagem

A figura 16 abaixo informa quais pinos do PIC foram utilizados no projeto analógico.

Figura 16 – Pinos utilizados no sistema analógico

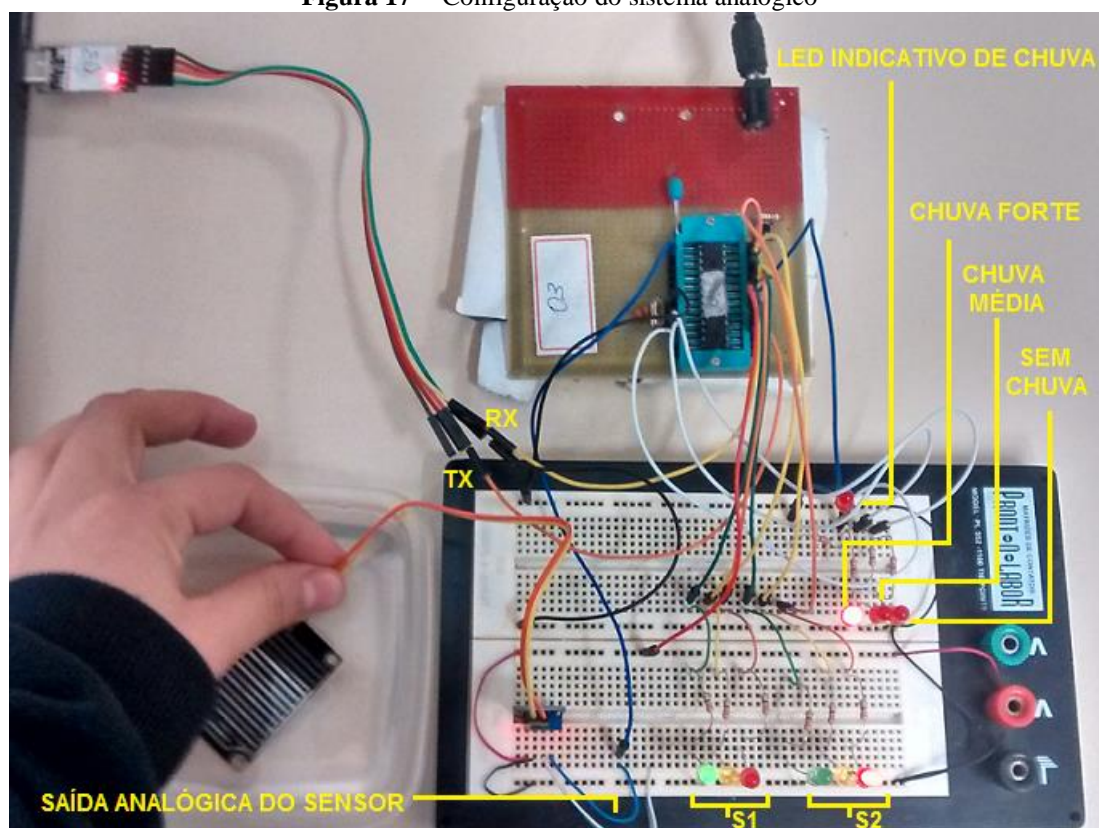


Fonte: própria autora (2016)

O pino GND foi conectado à fonte terra presente no *protoboard*; o pino VCC foi conectado ao *protoboard* para fornecer uma fonte de alimentação de 5 Volts para o sensor de chuva; o pino RB0 foi utilizado para enviar dados ao *LED* indicador de chuva; os pinos RB1 a RB6 foram utilizados para enviar dados aos *LEDs* verde (RB1), amarelo (RB2) e vermelho (RB3) do semáforo 1 e verde (RB4), amarelo (RB5) e vermelho (RB6) do semáforo 2; os pinos RC0, RC1 e RC2 foram utilizados, respectivamente, para indicar se naquele momento está sem chuva, com chuva média, ou com chuva forte; o pino AN0 recebe o valor analógico do sensor de chuva para posterior conversão A/D; já os pinos RX e TX são conectados ao cabo para envio e recebimento de dados por meio da comunicação serial.

O sistema montado ficará conforme a figura 17.

Figura 17 – Configuração do sistema analógico



Fonte: própria autora (2016)

3.3.2 Funcionamento

O sistema digital apresenta três ciclos: o ciclo sem chuva, o ciclo com chuva média e o ciclo com chuva forte.

O sinal analógico, após conversão, se torna um valor entre 0 e 1024. Quanto maior o valor, menos intensa a chuva. Valores próximos de 1024 indicam que não há chuva (esse valor pode variar caso o ambiente esteja úmido).

Consideremos C como o valor convertido. No sistema desenvolvido, está padronizado que:

- se $C \geq 768$, então não está chovendo;
- se $456 \leq C < 768$, então está chovendo com intensidade média;
- se $C < 456$, então está chovendo forte.

O ciclo sem chuva é temporizado da mesma forma que no sistema analógico: 10, 4 e 2 segundos para os sinais verde, amarelo e vermelho, respectivamente, para ambos os semáforos.

O ciclo com chuva média e o ciclo com chuva forte diferem-se do ciclo sem chuva quanto aos tempos dos sinais verdes. Para o ciclo com chuva média, o sinal verde do $s1$ permanece aberto durante 12 segundos e o do $s2$ durante 8 segundos. Já para o ciclo com chuva forte, o sinal verde do $s1$ permanece aberto durante 15 segundos e o do $s2$ durante 5 segundos.

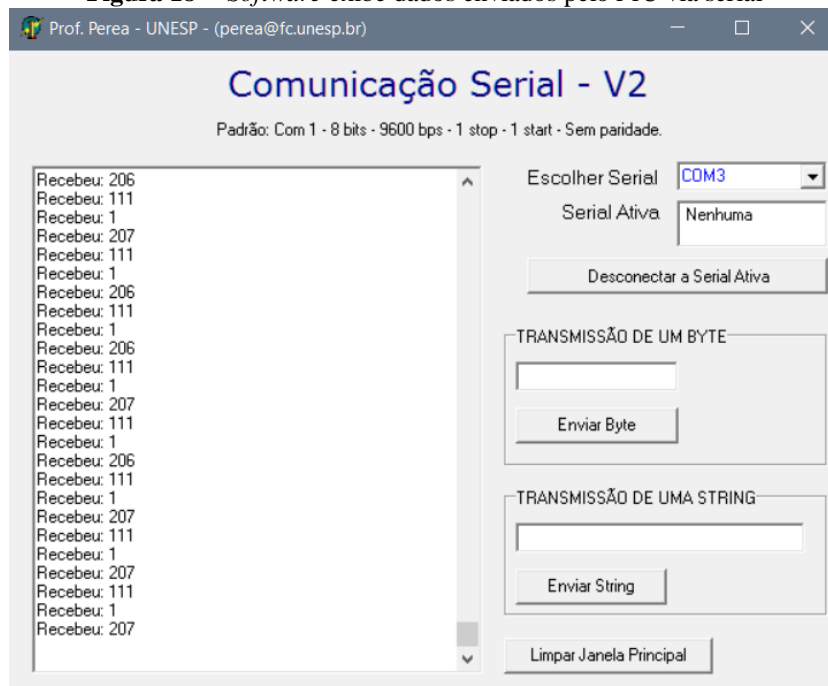
Além da variável que controla há quanto tempo está chovendo e do *LED* indicativo de chuva, cujo funcionamento encontra-se no tópico de funcionamento do sistema digital, o sistema apresenta também os *LEDs* que indicam a intensidade da chuva naquele momento (sem chuva / chuva média / chuva forte). Esses três *LEDs* são atualizados a cada segundo, de acordo com o valor pós conversão A/D em relação aos valores definidos anteriormente.

Para controle sobre qual ciclo irá executar, primeiramente testa-se se a variável indica que está chovendo há mais de 30 segundos, pois, se não estiver, o ciclo sem chuva é executado. Caso esteja, verifica-se qual *LED* está aceso, o de chuva média ou o de chuva forte, para, então, executar o ciclo correspondente ao *LED* aceso.

Essa sequência de testes é realizada infinitamente, permitindo a atualização da temporização dos semáforos de acordo com a ocorrência ou não de chuva e sua respectiva intensidade em determinado período.

Além disso, o sistema envia, via comunicação serial, os valores pós conversão A/D, após enviar o valor '111' para separar os dados de envios consecutivos. Utilizando um software para recepção de dados via serial, essas informações tornam-se visíveis, conforme ocorre no caso da figura 18.

Figura 18 — *Software* exibe dados enviados pelo *PIC* via serial



Fonte: própria autora (2016)

4 SOFTWARE COMPLEMENTAR

O sistema analógico requer um *software* complementar para visualização dos dados recebidos via serial. O *software* utilizado neste projeto foi o *COMUNICAÇÃO SERIAL – V2*, cedido pelo PROF. DR. JOÃO E. M. PEREA MARTINS. Sua interface consta na figura 18 já apresentada.

Além disso, foi desenvolvido, pela própria autora do projeto e na *IDE Lazarus* (versão 1.4.4), um sistema de contagem – chamado de *CONTADOR* – para otimizar a análise do funcionamento do projeto.

O sistema exibe na tela (em vermelho) o tempo decorrido a cada vez que a tecla *ENTER* é pressionada (exceto na primeira vez, a qual dá início à contagem), além do tempo decorrido desde o início da contagem (em azul). Possui também um botão para reiniciar a contagem. A interface encontra-se na figura 19 abaixo.

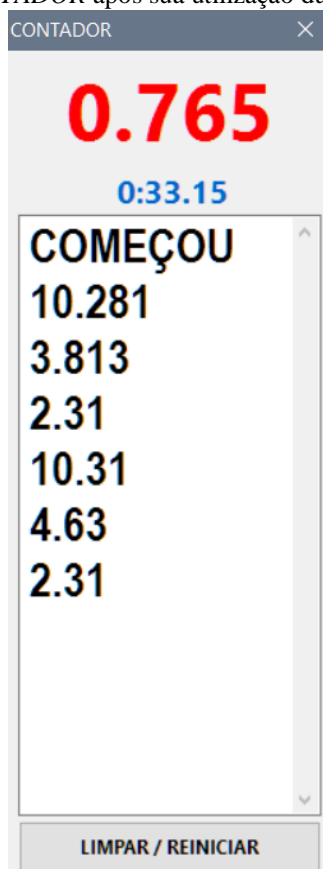
Figura 19 – Interface do sistema *CONTADOR*



Fonte: própria autora (2016)

Quando o projeto começava a executar, ao mesmo tempo a tecla *ENTER* era pressionada para iniciar a contagem. Toda vez que o semáforo mudava de cor, a tecla *ENTER* era novamente pressionada para a posterior visualização de quantos segundos a última cor ficou acesa. Por exemplo: o ciclo sem chuva apresenta, tanto para o semáforo 1 quanto para o semáforo 2, a seguinte temporização: 10 segundos com o sinal verde aceso, 4 segundos com o amarelo e 2 segundos com o vermelho (e enquanto o ciclo de um semáforo é executado, o outro semáforo mantém o sinal vermelho aceso). O *CONTADOR*, então, nos apresenta a interface apresentada na figura 20 após um ciclo de execução sem chuva, com os dados de tempo bastante aproximados de seus valores reais (10, 4, 2, 10, 4, 2).

Figura 20 – Interface do sistema *CONTADOR* após sua utilização durante a execução de um ciclo sem chuva



Fonte: própria autora (2016)

Esse software possibilitou uma análise mais eficiente do funcionamento do sistema.

BIBLIOGRAFIA

ARDUINO E CIA. **Ligando um sensor de chuva ao Arduino.** Disponível em:
<www.arduinoecia.com.br/2014/06/sensor-de-chuva-arduino.html>. Acesso em: 26 jun. 2016.

MICROCHIP. **PIC16F87X.** Disponível em:
<ww1.microchip.com/downloads/en/DeviceDoc/30292D.pdf>. Acesso em: 26 jun. 2016.