



UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FATECS – FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS
APLICADAS

CURSO DE ENGENHARIA DA COMPUTAÇÃO

DANIEL DOS SANTOS FERREIRA NETO

SISTEMA DE IRRIGAÇÃO AUTOMÁTICO

BRASÍLIA / DF
2º SEMESTRE DE 2008

DANIEL DOS SANTOS FERREIRA NETO

SISTEMA DE IRRIGAÇÃO AUTOMÁTICO

Monografia apresentada ao Curso de Engenharia da Computação, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: M.C Claudio Penedo

BRASÍLIA / DF
2º SEMESTRE DE 2008

Resumo

O principal objetivo deste trabalho foi desenvolver um modelo inteligente, com a utilização de tecnologia, hardware e software, com vistas a automatizar os procedimentos de irrigação de plantas, jardins e pequenas plantações, propiciando, em consequência, a redução no desperdício de água potável neste processo.

O interesse pelo estudo do assunto decorreu da necessidade de preservação do meio ambiente e dos recursos naturais disponíveis no planeta. No presente caso, se buscou a redução do consumo de água no processo de irrigação, com a sua utilização de forma automatizada.

O desenvolvimento do modelo permitiu a aplicação de conhecimentos e tecnologias para se produzir um sistema prático, automatizado, de fácil utilização, de baixo custo, que pode ser adotado na irrigação de residências e pequenas propriedades.

O modelo foi constituído por um hardware para emissão e outro para o recebimento do sinal via radiofrequência, acionado pelo software de forma manual ou automática quando solicitado pelo sensor de umidade, via porta paralela do computador.

Como esperado, o protótipo obtido demonstrou ser um sistema simples, que permite acionar e desligar, de forma automática e programada, o fornecimento de água no processo de irrigação.

Palavras-Chave: Irrigação, sensor de umidade, radiofrequência, porta paralela.

Abstract

The main goal of this study is to develop an intelligent model, using hardware and software technology to automate the procedures for irrigation of plants, gardens and small plantations, thus providing the reduction of water waste in the process.

The interest in this study ran from the need to preserve the environment and natural resources available on the planet. In this case, that aimed to reduce water consumption in the irrigation process, in an automated way.

This model allowed the application of knowledge and technology to produce a practical system, automated, easy to use, low cost, which can be used for irrigation of small homes and properties.

The model was developed using a device to send and another to receive the signal by radio frequency, driven by software, either manually or automatically when prompted by the humidity sensor, by the parallel port on the computer.

As expected, the prototype has shown to be a simple system, which allows to turn on and off the supply of water in the irrigation process in an automated and programmed way.

Keywords: irrigation, humidity sensor, radio frequency, parallel port.

*Dedico esta monografia a minha noiva,
Priscila dos Santos Fonseca, pelo
companheirismo, compreensão, carinho e
por sempre ter me motivado nos momentos
de dificuldade, nunca deixando que eu
desanimasse.*

Agradecimentos

Agradeço a minha família, em especial meus avós Daniel Ferreira e Myrthes Ferreira, por terem iniciado minha educação e meus pais, Daniel Ferreira Filho e Rosane B. Ferreira, por terem dado continuidade nesse árduo processo.

Agradeço ao meu irmão, Engenheiro Eurico B. Ferreira, pelo apoio durante todos os momentos na vida, pelos conselhos e pelas conversas, a você irmão, muito obrigado.

Ao meu sogro e amigo, Alcides Fonseca, pela dedicação, paciência e incentivo demonstrados.

A todos os amigos de faculdade que participaram dessa longa jornada, em especial Gustavo de Faria (Capitão), Roberta Neder (Beta), Keyth Roy e Danilo Resende.

Sumário

1	INTRODUÇÃO	8
1.1	Contextualização do Trabalho	8
1.2	Motivação.....	9
1.3	Objetivo do Projeto.....	9
1.4	Estrutura do Trabalho	10
2	CONCEITOS SOBRE SISTEMAS DE IRRIGAÇÃO	12
2.1	Irrigação	12
2.2	Métodos de irrigação.....	13
2.3	Sistemas de irrigação.....	13
2.3.1	Gotejamento	13
2.3.2	Aspersão Convencional	15
2.3.3	Inundação	15
2.3.4	Pivô Central	16
3	DESENVOLVIMENTO DO HARDWARE E DO SOFTWARE	17
3.1	DESCRIÇÃO DO HARDWARE.....	17
3.1.1	Porta Paralela	17
3.1.2	Resistores.....	22
3.1.3	Capacitores.....	22
3.1.4	Circuito Integrado MC 145026	23
3.1.5	Circuito Integrado MC 145027	24
3.1.6	Circuito Integrado 74LS244	26
3.1.7	Emissor RF TX – C1	27
3.1.8	Conversor A/D	29
3.1.9	Sensor de Umidade	30
3.2	PROCESSO DE CONSTRUÇÃO DO HARDWARE	31
3.2.1	Teste de oscilação via cabo de dados	31
3.2.2	Teste de oscilação via Radiofrequência	35
3.2.3	Integração dos componentes.....	38
3.3	SOFTWARE.....	42
3.3.1	Tecnologia Utilizada.....	42
3.3.2	Fluxograma	42
3.3.3	Funcionamento	43

4	RESULTADOS OBTIDOS.....	45
4.1	Acionamento manual do sistema	45
4.2	Acionamento Automático do Sistema	47
4.3	Dificuldades Encontradas	51
5	CONCLUSÃO	52
	REFERÊNCIAS.....	54
	ANEXO I – DATASHEET DOS COMPONENTES UTILIZADOS NO HARDWARE ..	55
	ANEXO II – CÓDIGO FONTE DO SOFTWARE.....	68

Índice de Figuras

Figura 1.1 – Visão geral do projeto.	10
Figura 2.1 – Modelo de Gotejador.....	14
Figura 2.2 – Irrigação por gotejamento.	14
Figura 2.3 – Irrigação por Aspersão.	15
Figura 2.4 – Irrigação por Inundação.	16
Figura 2.5 – Irrigação com Pivô.....	16
Figura 3.1 – Comunicação Serial.	18
Figura 3.2 – Comunicação Paralela.	19
Figura 3.3 – Identificação dos pinos.....	21
Figura 3.4 – MC 145026.....	23
Figura 3.5 – Diagrama de blocos MC 145026.....	24
Figura 3.6 – MC 145027.....	25
Figura 3.7 – Diagrama de blocos MC 145027.....	26
Figura 3.8 – CI 74LS244.	26
Figura 3.9 – Diagrama de blocos CI 74LS244.....	27
Figura 3.10 – TX-C1.....	27
Figura 3.11 – RX-D1.	28
Figura 3.12 – Conversor A/D.....	29
Figura 3.13 – Diagrama de blocos Conversor A/D.....	30
Figura 3.14 – Sensor de umidade.	30
Figura 3.15 – Endereços encoder e decoder.	33
Figura 3.16 – Transmissão via cabo de dados.....	33
Figura 3.17 – Validação da transmissão via cabo.....	34
Figura 3.18 – Endereços na Transmissão RF.....	35
Figura 3.19 – Transmissão via RF.	36
Figura 3.20 – Validação da transmissão via RF.....	37
Figura 3.21 – Esquema elétrico do solenóide.	39
Figura 3.22 – Esquema elétrico porta paralela.....	40
Figura 3.23 – Esquema elétrico do hardware completo.	40
Figura 3.24 – Hardware completo.	41
Figura 3.25 – Fluxograma do software.....	43

Figura 4.1 – Acionamento manual.....	45
Figura 4.2 – Leds acessos e solenóide aberto.	46
Figura 4.3 – Desligamento via software.	46
Figura 4.4 – Leds desligados e solenóide fechado.	47
Figura 4.5 – Valor da umidade detectado pelo sensor: primeira medição.....	48
Figura 4.6 – Valor da umidade detectado pelo sensor: segunda medição.....	48
Figura 4.7 – Acionamento do sistema pelo sensor.....	49
Figura 4.8 – Valor da umidade detectado pelo sensor: terceira medição.....	50
Figura 4.9 – Desativação do sistema pelo sensor.....	51

Índice de Tabelas

Tabela 3.1 – Endereços das Portas	19
Tabela 3.2 – Registradores	20
Tabela 3.3 – Pinagem do conector DB25.....	22
Tabela 3.4 – Pinos Emissor.....	28
Tabela 3.5 – Pinos Receptor.....	29
Tabela 3.6 – Endereçamento em modo binário.....	32
Tabela 3.7 – Descrição dos itens da Figura 4.9	41

1 INTRODUÇÃO

1.1 Contextualização do Trabalho

Uma das grandes preocupações atuais é a preservação do meio ambiente e dos recursos naturais disponíveis no planeta.

O uso indiscriminado e inconseqüente desses recursos pelo homem no decorrer de décadas trouxe sérias conseqüências que vêm afetando as condições de vida em nosso mundo, trazendo previsões catastróficas a respeito do futuro do planeta se atitudes e providências não forem tomadas.

Entretanto, um fato animador é o aumento da consciência dos povos e de seus governantes quanto à gravidade da situação, o que vem possibilitando a adoção de inúmeras medidas para conter os abusos e proporcionar um futuro melhor para a humanidade.

Dentre os vários recursos naturais, um que merece atenção especial é a água. Tendo em vista que ela é imprescindível para a vida de todos os seres da Terra, é fundamental que a sua utilização seja feita de forma adequada e responsável para que esse recurso vital esteja sempre disponível.

A água é usada de muitas maneiras e para diversas finalidades, destacando-se a sua utilização para irrigação residencial de plantas, jardins e, também, em áreas cultivadas nas pequenas propriedades. Nota-se, mesmo numa observação superficial que, nesses casos, o uso da água é inadequado, principalmente no que se refere ao desperdício do recurso pela forma pouco metódica, manual e sem automação pelo qual é feito.

Se considerado isoladamente, tal desperdício pode não parecer significativo, mas diante da grande quantidade de usuários, o resultado final provocado pode trazer um considerável desperdício de água, que deve ser evitado.

1.2 Motivação

O assunto provoca um forte interesse em utilizar os conhecimentos e a tecnologia para se produzir um sistema prático, automatizado, de fácil utilização, de baixo custo, que possa ser adotado na irrigação de residências e pequenas propriedades.

O modelo produzido poderá ser aplicável e útil para milhares de pessoas que necessitam e usam algum sistema de irrigação de pequeno porte, trazendo melhorias no aproveitamento adequado e na economia de uso da água para tal finalidade.

1.3 Objetivo do Projeto

O principal objetivo deste projeto é desenvolver um modelo inteligente, com a implementação de hardware e software, com vistas a automatizar os procedimentos de irrigação de plantas, jardins e pequenas plantações, propiciando, em consequência, a redução no desperdício de água potável neste processo.

O modelo desenvolvido possibilita ainda:

- O controle e diminuição do desperdício da água, protegendo o meio ambiente e reduzindo custos operacionais;
- A irrigação das plantas com a quantidade adequada de água melhorando a qualidade do produto final (frutas, verduras, legumes, etc.)
- A utilização de tecnologias a favor do homem como um facilitador do trabalho;
- A eliminação da necessidade do deslocamento humano para o acionamento/desligamento manual do sistema de irrigação.

- O controle automático do processo de irrigação com o uso do computador, com o acionamento do sistema por radiofrequência, minimizando a ação do homem junto à plantação.

Para melhor entendimento do trabalho desenvolvido, a Figura 1.1 apresenta uma visão geral do projeto.

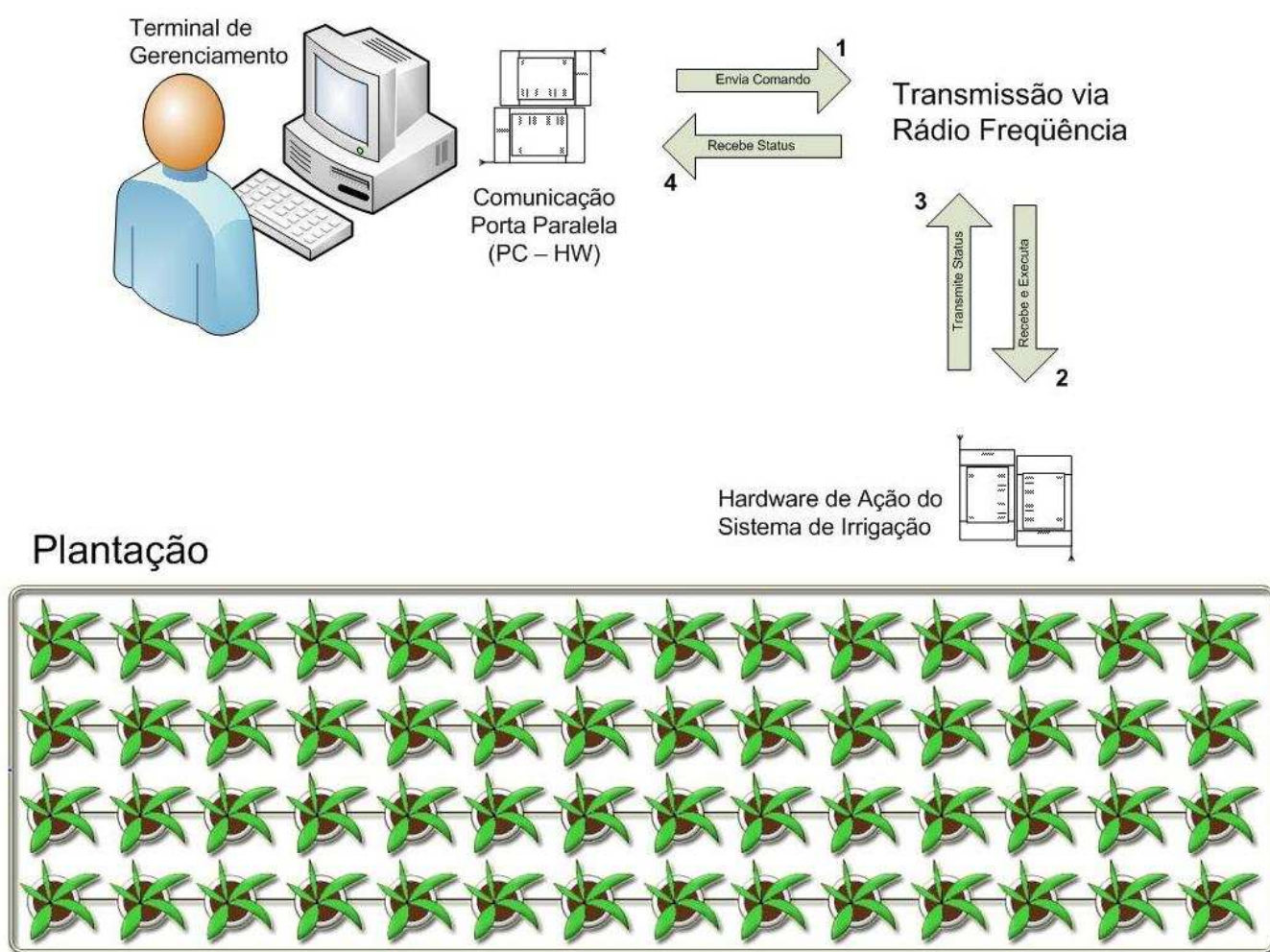


Figura 1.1 – Visão geral do projeto.

1.4 Estrutura do Trabalho

Além deste capítulo introdutório e da conclusão, o trabalho está estruturado em mais quatro capítulos, a seguir descritos:

No Capítulo 2 – Conceitos sobre sistemas de irrigação – são apresentados conceitos teóricos sobre as diversas técnicas, sistemas e métodos de irrigação existentes.

No Capítulo 3 – Desenvolvimento do Hardware e do Software - são descritos os componentes eletrônicos utilizados, os processos de construção dos hardwares e do software, e o funcionamento do modelo proposto.

No Capítulo 4 - Resultados Obtidos – são apresentados os resultados obtidos neste projeto.

2 CONCEITOS SOBRE SISTEMAS DE IRRIGAÇÃO

Para melhor compreensão do trabalho, neste capítulo são mostrados os conceitos sobre irrigação e alguns dos sistemas e métodos utilizados na agricultura.

Em princípio, o modelo desenvolvido pode ser utilizado em diversos sistemas de irrigação. Entretanto, por tratar-se de um protótipo acadêmico, o mesmo foi desenvolvido até a etapa do acionamento do dispositivo que permite a passagem/interrupção no fluxo da água, não sendo realizados testes com qualquer sistema de irrigação.

2.1 Irrigação

A irrigação é definida como um conjunto de técnicas e equipamentos, programados e operados de forma racional, que têm o objetivo principal de repor a água do solo, suprimindo a falta, insuficiência ou má distribuição das chuvas, proporcionando, desta forma, a umidade necessária ao crescimento normal das plantas nele existentes.

Trata-se de uma técnica bastante utilizada na agricultura que garante o fornecimento de água para as plantas, em quantidade suficiente e no momento certo, assegurando uma melhor qualidade no processo agrícola e, até mesmo, a própria sobrevivência da plantação.

Essa técnica, quando bem planejada e colocada em operação, proporciona as condições mais adequadas ao bom desenvolvimento da plantação, aumentando a sua produtividade e trazendo, por conseguinte, maiores lucros para o produtor.

As plantas possuem cinco necessidades básicas: local para crescer, temperatura ideal, ar, luz e água. Dentre elas, a água merece destaque, pois, além de ser o responsável pelo transporte dos nutrientes encontrados no solo, que são fundamentais para o crescimento e desenvolvimento das plantas, possui um papel muito importante na germinação e no processo de fotossíntese.

2.2 Métodos de irrigação

O método de irrigação é a forma pela qual a água é disponibilizada para as plantas em geral. Existem basicamente quatro métodos:

- *Superfície*: a distribuição se dá devido à gravidade através da superfície do solo.
- *Aspersão*: jatos de água são lançados para o alto e caem sobre as plantas em forma de chuva.
- *Localizada*: a água é aplicada em apenas uma fração do sistema radicular das plantas (coroas).
- *Subirrigação*: a água é conduzida sob pressão por tubos e aplicada de forma pontual com pequena intensidade na superfície do solo.

Para cada método existem dois ou mais sistemas de irrigação que podem ser empregados. A escolha dependerá de vários fatores, tais como a declividade do terreno, o tipo de solo, a necessidade da plantação, o clima local, a precipitação natural, etc.

2.3 Sistemas de irrigação

Trata-se do conjunto de ações e equipamentos necessários para se colocar em prática o método de irrigação escolhido para determinada plantação. Os sistemas de irrigação mais adotados são: Gotejamento, Aspersão Convencional, Inundação e Pivô Central, descritos a seguir.

2.3.1 Gotejamento

A irrigação por gotejamento é baseada na pequena vazão através de orifícios de diâmetros reduzidos, situados em estruturas especiais denominadas gotejadores, adaptadas em tubulações de plásticos localizados próximos as plantas.

Na Figura 2.1 pode-se verificar um modelo de gotejador existente.



Figura 2.1 – Modelo de Gotejador.

Nesse gotejador é possível determinar a vazão de água que será utilizada para a irrigação da planta. Essa vazão é determinada pela rotação da peça central permitindo com que aumente ou diminua o fluxo de água.

A Figura 2.2 ilustra uma área maior sendo irrigada com os gotejadores.



Figura 2.2 – Irrigação por gotejamento.

Nota-se um melhor aproveitamento da água com esse sistema, eliminando o fornecimento de água em áreas que não seriam aproveitadas pelas plantas.

2.3.2 Aspersão Convencional

Na irrigação por aspersão a água é aplicada na plantação em forma de chuva artificial através do fracionamento do jato em um número enorme de gotas de águas que se espalham pelo ar, caindo sobre o terreno. Esse fracionamento se dá devido ao fluxo de água sobre pressão através de pequenos orifícios, sendo ilustrada na Figura 2.3.



Figura 2.3 – Irrigação por Aspersão.

Observa-se nesse sistema que o fornecimento da água é uniforme, ou seja, ela é distribuída dentro de uma circunferência limitada pelo tipo e qualidade do aspersor.

2.3.3 Inundação

O sistema de irrigação por inundação consiste no fornecimento da água ao terreno, aplicando-se água diretamente nos sulcos vizinhos à superfície onde se encontra a plantação. Com isso, cria-se uma lâmina de água mostrada na Figura 2.4.



Figura 2.4 – Irrigação por Inundação.

Percebe-se um grande volume de água percorrendo os sulcos. Esse sistema de irrigação é bastante utilizado em culturas que necessitam de bastante água durante longos períodos.

2.3.4 Pivô Central

Consiste em uma tubulação de grande porte com vários aspersores ou bocais, devidamente espaçados, suspensa acima da plantação por meio de pequenas torres que possuem rodas e dispositivo motor que gira em torno de um eixo enquanto irriga o terreno. A Figura 2.5 ilustra um modelo existente de pivô.



Figura 2.5 – Irrigação com Pivô.

3 DESENVOLVIMENTO DO HARDWARE E DO SOFTWARE

O modelo é desenvolvido com a utilização de um hardware para emissão e outro para o recebimento do sinal via radiofrequência, acionado pelo software de forma manual ou automática quando solicitado pelo sensor de umidade, via porta paralela do computador.

Neste capítulo são descritos os componentes eletrônicos utilizados e suas características, os processos de construção dos hardwares e do software de controle, bem como o funcionamento do modelo proposto.

3.1 DESCRIÇÃO DO HARDWARE

São utilizados para a construção desse projeto os seguintes componentes: porta paralela, resistores, capacitores, circuitos integrados, emissores e receptores de radiofrequência, conversor A/D e sensor de umidade.

3.1.1 Porta Paralela

A porta paralela é uma interface presente nos computadores. Sua utilização permite realizar a comunicação entre o computador e outros periféricos. A porta paralela adota o padrão Transistor Transistor Logic (TTL) nos seus pinos, de forma que um pino esteja em nível lógico baixo (0) quando sua tensão elétrica estiver entre 0 e 0,4 V ou esteja em nível lógico alto (1) quando a tensão estiver entre 3,1 e 5 V.

Neste projeto, a porta paralela faz a comunicação entre o hardware de acionamento do sistema de irrigação com o computador, além de ser utilizada para efetuar a leitura do sensor de umidade.

A seguir são tratados aspectos relativos a transmissão de dados, endereçamento, registradores e conector DB25, inerentes ao funcionamento da porta paralela.

3.1.1.1 Transmissão de dados

Este trabalho faz uso das duas formas de transmissão de dados utilizadas pela porta paralela: a transmissão serial, usada no envio dos bits de endereços e dados do encoder para o decoder, e a transmissão paralela, usada na comunicação do software com o hardware.

3.1.1.1.1 Transmissão Serial

A transmissão serial é dada pelo envio de dados em um fluxo através de um canal, enviando bit a bit, ou seja, enfileirados, onde cada bit é seguido de outro bit, conforme Figura 3.1.



Figura 3.1 – Comunicação Serial.

Nota-se que os bits são enviados na ordem em que são solicitados. Na Figura 3.1 são mostrados os bits com uma seqüência de 1 a 8 para uma melhor visualização da transmissão.

3.1.1.1.2 Transmissão Paralela

A transmissão paralela possui como característica o envio simultâneo de um bit em cada canal, totalizando 8 bits de dados transmitidos simultaneamente, conforme ilustrado na Figura 3.2.

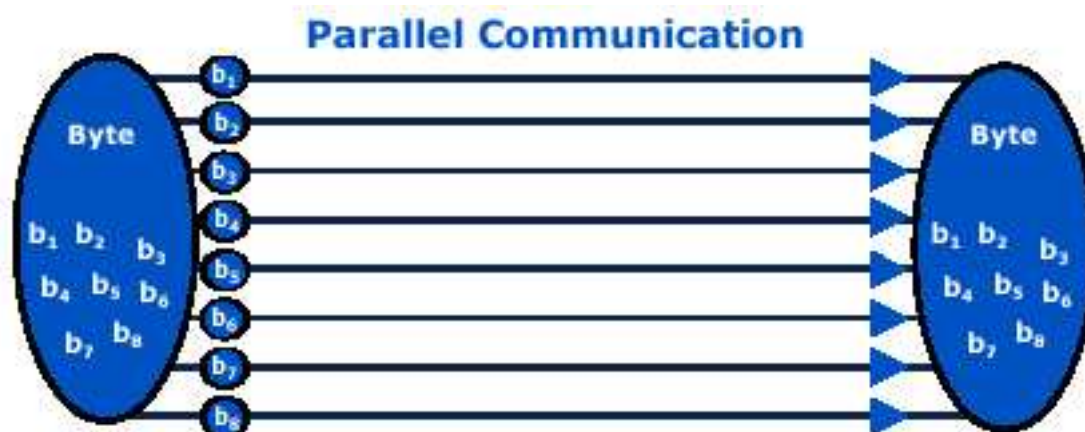


Figura 3.2 – Comunicação Paralela.

Percebe-se que os bits são enviados em canais separados. Na Figura 3.2 são mostrados os bits numa sequência de 1 a 8 para uma melhor visualização da transmissão que ocorre ao mesmo tempo para todos os bits.

3.1.1.2 Endereçamento

O computador nomeia as portas paralelas de LTP1, LTP2, LTP3, etc. Pode-se efetuar a leitura dos dados utilizando a porta paralela, mas para isso é necessário conhecer seus endereços lógicos. Por padrão são: 378h para LTP1 e 278h para LTP2, ilustrado na Tabela 3.1.

Tabela 3.1 – Endereços das Portas

Nome da Porta	Endereço de memória	Endereço da Porta	
		Hexadecimal	Decimal
LTP1	0000:0408	378	888
LTP2	0000:040A	278	632

3.1.1.3 Registradores

A porta paralela possui três registradores: o de dados, o de status e o de controle. O registrador de dados é utilizado para o envio de um byte de dados; o

registrador de status é usado para receber dados externos e o registrador de controle para dados de controle, conforme descrito na Tabela 3.2.

Tabela 3.2 – Registradores

Nome	Endereço LTP1	Endereço LTP2	Descrição
Registro de Dados	378h	278h	Envia um byte
Registro de Status	379h	279h	Ler o status
Registro de Controle	37Ah	27Ah	Envia dados de controle

Ao se efetuar a leitura no endereço de status (379h) é possível obter as informações referentes aos pinos: *busy*, *acknowledge*, *Paper*, *Select* e *Error* encontrados na porta paralela.

A escrita de dados no endereço 37Ah permite o controle dos dispositivos que estão disponíveis na porta paralela.

3.1.1.4 Conector DB25

O conector DB25 está localizado na parte posterior do gabinete do computador. É através dele que o cabo paralelo se conecta ao computador com a função de enviar e receber os dados. Cada pino é identificado pela sua posição e característica.

Ele possui 25 pinos utilizados para registro de dados(D), registro de status(S) ou registro de controle(C), conforme Figura 3.3.

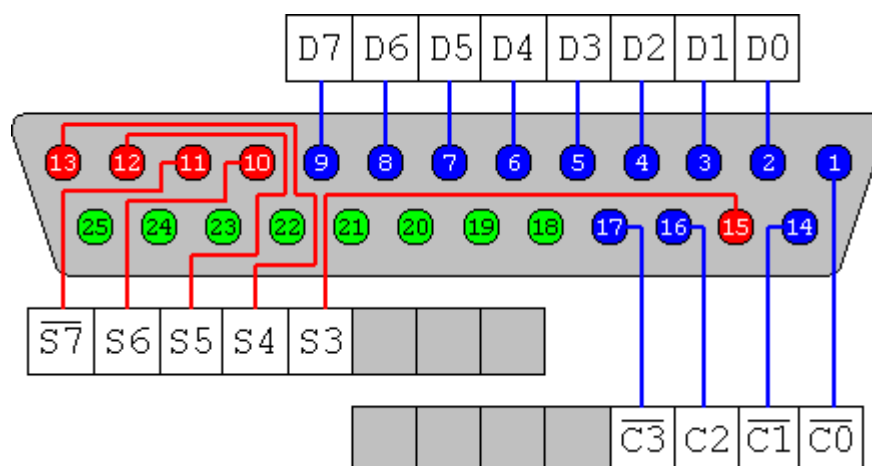


Figura 3.3 – Identificação dos pinos.

- Os pinos 2 a 9, são pinos de DADOS. Sua função é enviar os dados via porta paralela.
- Os pinos 10,11,12,13 e 15 são pinos de STATUS. São responsáveis pela troca de mensagens fornecendo o respectivo status.
- Os pinos 1,14,16 e 17 são pinos de CONTROLE. São usados para fazer a interface e a troca de mensagens entre o computador e o dispositivo ligado a porta paralela.
- Os pinos 18 a 25 são pinos usados para o aterramento.

Existem alguns pinos na porta paralela que trabalham com o conceito de lógica invertida e são identificados com uma barra deitada acima do código do pino ($\overline{S7}$, $\overline{C3}$, $\overline{C1}$ e $\overline{C0}$). Para se ativar um desses pinos será necessário enviar o sinal lógico “0” e para desativá-lo será necessário enviar o sinal lógico “1”.

Na Tabela 3.3, pode-se verificar as informações pertinentes a cada pino:

Tabela 3.3 – Pinagem do conector DB25

Pin Nº (DB25)	Nome	I/O	Tipo Registro	Invertido?
1	Strobe	Out	Controlo-0	Sim
2	Data0	In/Out	Data-0	Não
3	Data1	In/Out	Data-1	Não
4	Data2	In/Out	Data-2	Não
5	Data3	In/Out	Data-3	Não
6	Data4	In/Out	Data-4	Não
7	Data5	In/Out	Data-5	Não
8	Data6	In/Out	Data-6	Não
9	Data7	In/Out	Data-7	Não
10	Ack	In	Status-6	Não
11	Busy	In	Status-7	Sim
12	Paper-Out	In	Status-5	Não
13	Select	In	Status-4	Não
14	Linefeed	Out	Controlo-1	Sim
15	Error	In	Status-3	Não
16	Initialize	Out	Controlo-2	Não
17	Select-Printer	Out	Controlo-3	Sim
18-25	Ground	-	-	-

3.1.2 Resistores

São utilizados resistores de três faixas na construção desse projeto, com a finalidade de oferecer resistência às passagens de corrente elétrica, limitando sua intensidade e transformando energia elétrica em energia térmica.

3.1.3 Capacitores

No projeto são utilizados capacitores de cerâmica, que têm a função principal de armazenar energia elétrica, funcionando como um filtro, deixando passar a corrente alternada e obstruindo a passagem da corrente contínua.

3.1.4 Circuito Integrado MC 145026

Para codificar os sinais de entrada é utilizado o circuito integrado MC145026. Ele possibilita que o receptor interprete somente os sinais codificados enviados pelo transmissor. Isto evita que um conjunto receba o sinal transmitido por outro. Mesmo que possa ser captado, ele não poderá ser compreendido. Com isto, garante-se que vários equipamentos podem ser utilizados num mesmo local sem que um interfira no outro.

Esta codificação é realizada conforme a posição dos pinos de endereço (pinos A1, A2, A3, A4 e A5) e de dados (D6, D7, D8 e D9). Esses pinos podem estar ligados de dois modos, binário (ligados ao terra ou a alimentação) ou trinário (ligados ao terra, a alimentação ou em aberto, não ligados a nada).

No modo trinário pode-se obter até 19.683 ($3^9 = 19.683$) combinações no circuito MC 145026, utilizando os cinco pinos de endereços (A1, A2, A3, A4 e A5) com os quatro pinos de dados (D6, D7, D8 e D9). Já no modo binário pode-se obter até 512 combinações ($2^9 = 512$).

Além disso, ele necessita estar ligado a componentes externos para definir o valor da frequência. Por se tratar de componentes analógicos, a gama de valores dessa oscilação é muito grande, oferecendo assim um recurso que ajuda a proteger o circuito contra interferências de outros sistemas que estejam operando na mesma frequência. A Figura 3.4 apresenta o circuito integrado MC 145026.

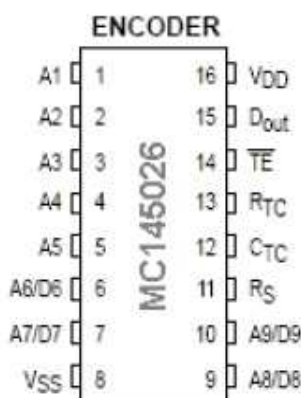


Figura 3.4 – MC 145026.

O diagrama de blocos contém informações pertinentes a lógica booleana deste circuito e pode ser observado na figura 3.5.

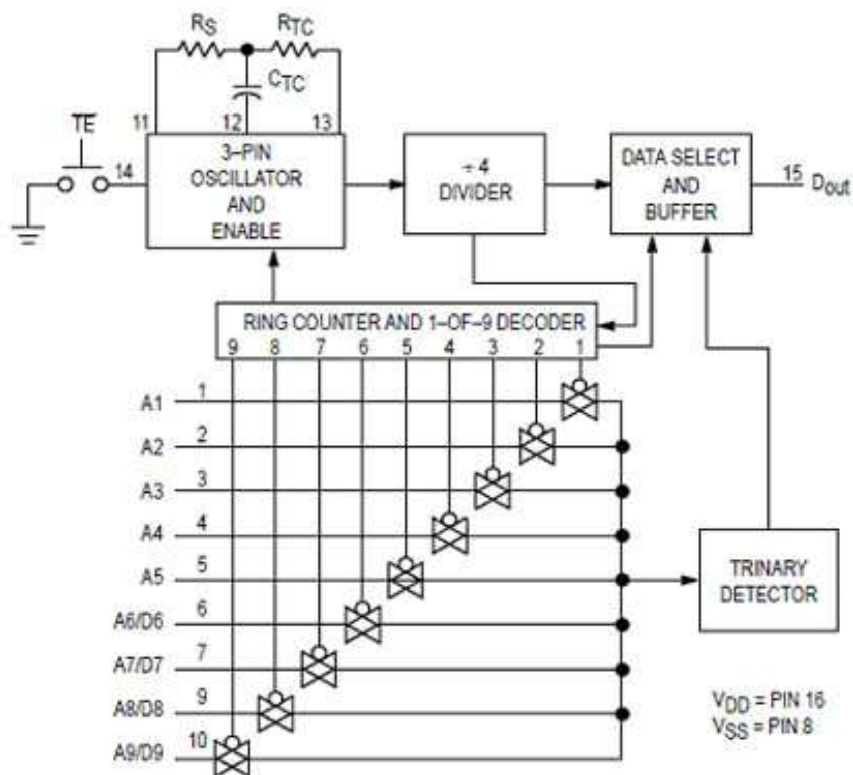


Figura 3.5 – Diagrama de blocos MC 145026

3.1.5 Circuito Integrado MC 145027

Para decodificar os sinais recebidos pelo encoder é utilizado o circuito integrado MC 145027. Ele decodifica, interpreta e executa os sinais que são enviados pelo encoder. A decodificação significa receber um código e transformá-lo em informações compreensíveis e úteis.

O circuito integrado MC 145027 trabalha entre as tensões 4,5V e 18V e possui 16 terminais, onde cada um possui uma função. A Figura 3.6 ilustra este tipo de circuito.

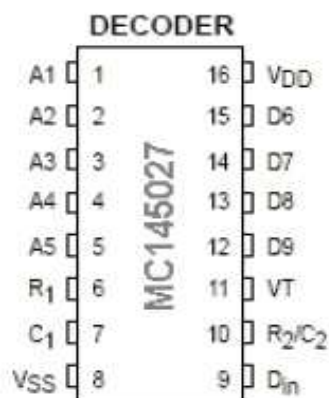


Figura 3.6 – MC 145027.

Esta decodificação é realizada conforme a posição dos pinos de endereço (pinos A1, A2, A3, A4 e A5) e de dados (D6, D7, D8 e D9). Esses pinos podem estar ligados de dois modos, binário (ligados ao terra ou a alimentação) ou trinário (ligados ao terra, a alimentação ou em aberto, não ligados a nada).

No modo trinário pode-se obter até 19.683 ($3^9 = 19.683$) combinações no circuito MC 145026, utilizando os cinco pinos de endereços (A1, A2, A3, A4 e A5) com os quatro pinos de dados (D6, D7, D8 e D9). Já no modo binário pode-se obter até 512 combinações ($2^9 = 512$).

Os pinos de endereços do decoder (pinos A1, A2, A3, A4 e A5) devem estar configurados conforme os endereços no encoder. Caso os endereços estejam configurados de maneira distinta, não há comunicação entre o encoder e decoder. Porém, se o par de encoder/decoder estiver com a mesma configuração de endereços, o pino 11 (VT – Valid Transmission) do decoder será ativado indicando que as informações na saída dos pinos de dados foram atualizadas.

O diagrama de blocos contém informações pertinentes a lógica booleana deste circuito e pode ser observado na figura 3.7.

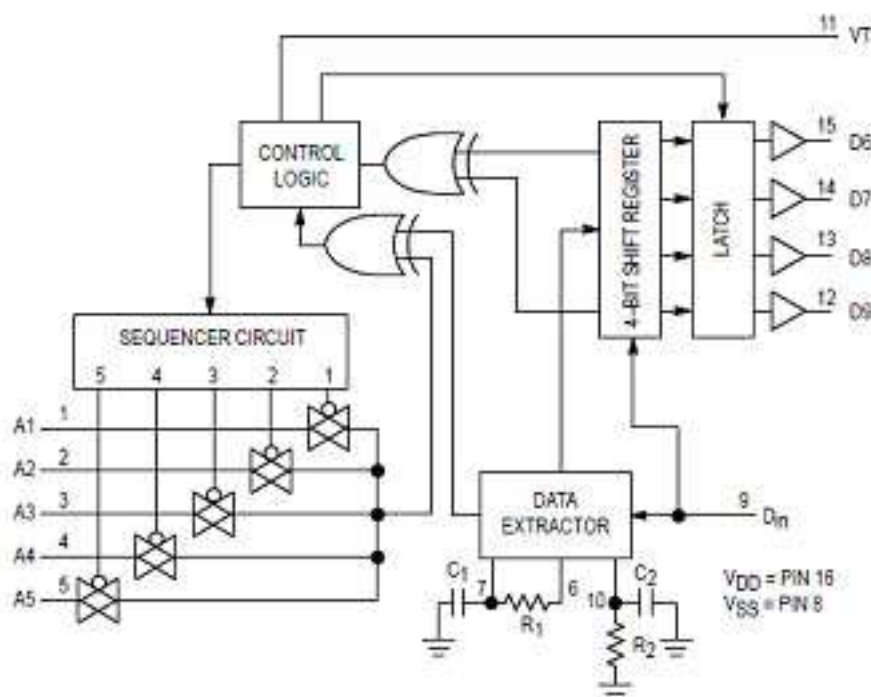


Figura 3.7 – Diagrama de blocos MC 145027

3.1.6 Circuito Integrado 74LS244

Para comunicação do hardware com o computador foi usado o circuito integrado 74LS244 como buffer. O buffer possui uma área reservada de memória para armazenar os endereços e aumentar a velocidade de acesso aos dados utilizados por um dispositivo, seja hardware e/ou software. Na Figura 3.8 este buffer está representado.

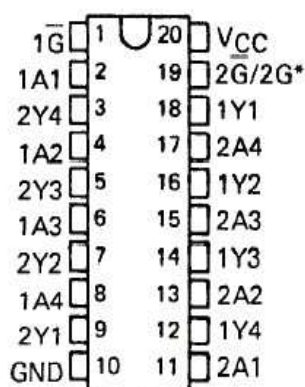


Figura 3.8 – CI 74LS244.

Esse circuito funciona com a lógica booleana representada no diagrama de blocos ilustrado na Figura 3.9.

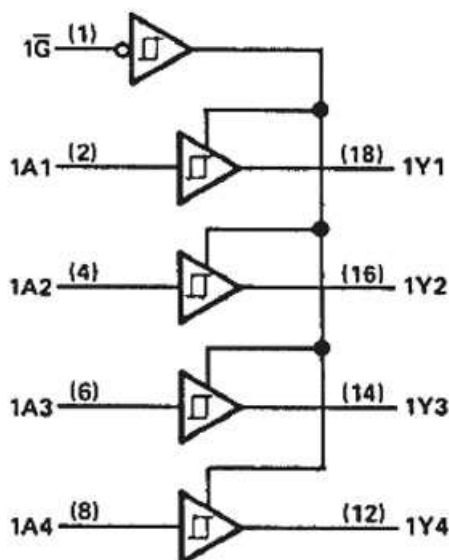


Figura 3.9 – Diagrama de blocos CI 74LS244

O buffer (TTL) 74LS244 serve também de proteção para a porta paralela do computador em caso de sobrecarga de tensão.

3.1.7 Emissor RF TX – C1

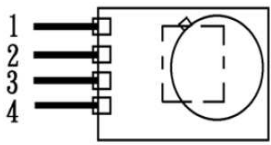
Neste trabalho é utilizado o emissor RF TX-C1, nas freqüências de 315 MHz e 433,9 MHz, que permite realizar a troca de informações por meio de radiofreqüência, sendo responsável pelo envio dos bits do MC 145026 (encoder) para o receptor que está junto ao MC 145027 (decoder). Este emissor é apresentado na Figura 3.10.



Figura 3.10 – TX-C1.

O transmissor possui 04 (quatro) pinos e suas funções estão detalhadas na Tabela 3.4:

Tabela 3.4 – Pinos Emissor

Pino	Função	Desenho
1	GND	
2	DATA	
3	VCC	
4	ANT	

3.1.7.1 Receptor RF RX – D1

O projeto utiliza o receptor RF RX-D1, nas frequências de 315 MHz e 433,9 MHz, responsável pelo recebimento dos bits enviados pelo emissor. A Figura 3.11 ilustra o receptor.



Figura 3.11 – RX-D1.

Esse componente recebe os bits via radiofrequência e os entrega para o MC 145027 (decoder) para verificação dos endereços. Essa verificação, sendo positiva, faz com que o decoder execute as solicitações enviadas, sejam elas de ligar ou desligar qualquer dispositivo.

Para que haja comunicação entre os componentes (emissor e receptor), os mesmos devem estar trabalhando na mesma frequência. Ambos os módulos podem ser alimentados com tensões entre 2V e 14V.

O receptor possui 08 (oito) pinos e suas funções estão detalhadas na Tabela 3.5:

Tabela 3.5 – Pinos Receptor

Pino	Função	Desenho
1	ANT	
2	GND	
3	GND	
4	VCC	
5	VCC	
6	DATA	
7	DATA	
8	GND	

3.1.8 Conversor A/D

Para converter grandezas analógicas em grandezas digitais foi utilizado o conversor A/D, ilustrado na Figura 3.12.

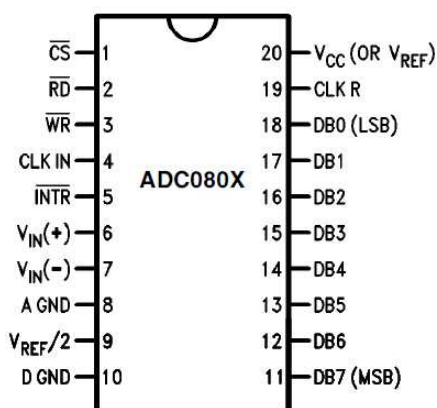


Figura 3.12 – Conversor A/D.

O processo de conversão analógico/digital é feito de modo paralelo, com todos os bits processados ao mesmo tempo, possibilitando uma maior velocidade na operação.

Este conversor funciona com a lógica booleana representada no diagrama de blocos observado na Figura 3.13.

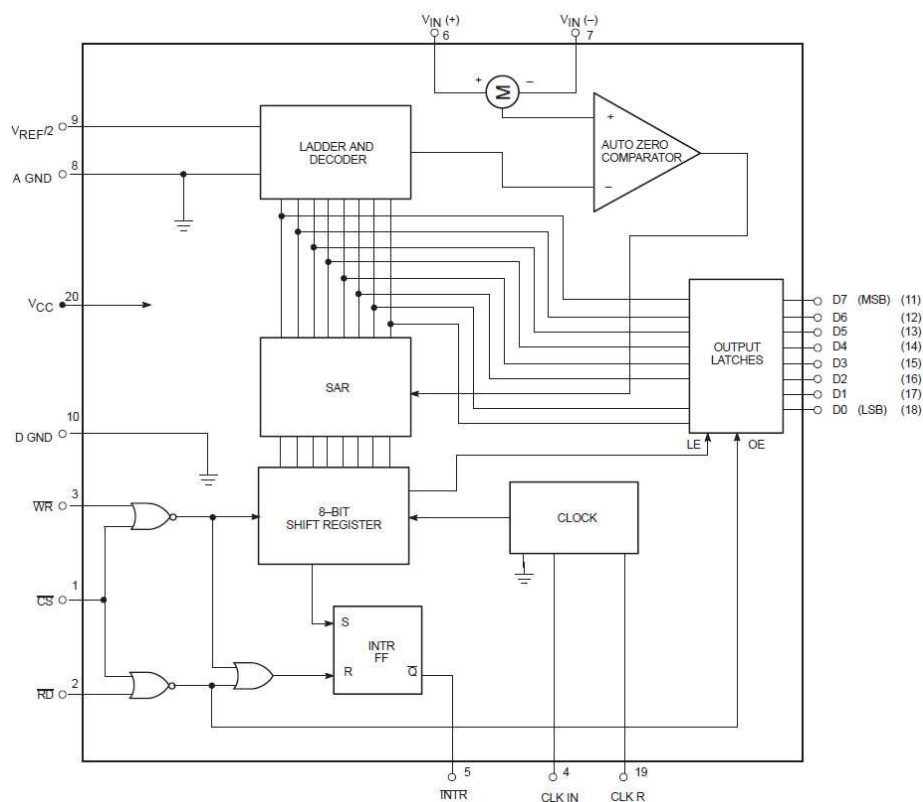


Figura 3.13 – Diagrama de blocos Conversor A/D

3.1.9 Sensor de Umidade

Para detectar a diferença de potencial elétrico entre os terminais expostos ao solo é utilizado o sensor de umidade, que é constituído por uma placa de circuito impresso ilustrado na Figura 3.14.

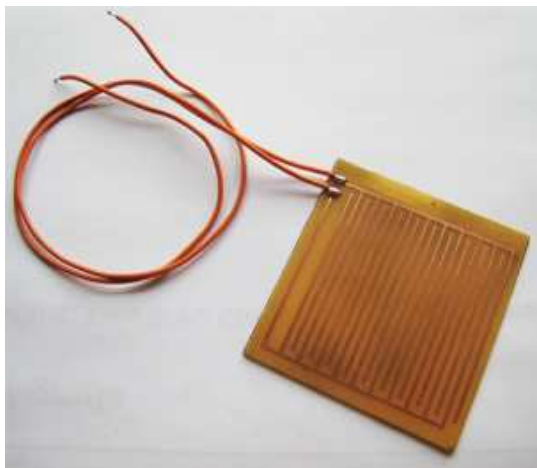


Figura 3.14 – Sensor de umidade.

A medição da diferença do potencial elétrico entre os terminais expostos ao solo é feita por intermédio dos sais minerais presentes na solução (solo + água), que auxiliam no transporte de elétrons entre os terminais do sensor.

3.2 PROCESSO DE CONSTRUÇÃO DO HARDWARE

O hardware é desenvolvido de forma gradativa. Os componentes são progressivamente adicionados ao projeto na medida em que são validados nos testes efetuados.

Os testes são realizados em três etapas distintas. A primeira delas contempla o funcionamento de controle (encoder/decoder) pela transmissão via cabo. Na segunda etapa ocorre a troca do meio de comunicação – substituição do cabo por módulos de radiofrequência. Na terceira etapa são adicionados os demais componentes do protótipo (sensor de umidade, solenóide e porta paralela), utilizando a transmissão por radiofrequência.

3.2.1 Teste de oscilação via cabo de dados

Nesta etapa é verificada a comunicação entre o encoder e o decoder por meio da ligação direta com o cabo de dados, para constatação do correto funcionamento dos módulos de controle.

Os circuitos integrados MC 145026 e MC 145027 possuem dois portadores diferentes de dados, sendo 5 portas utilizadas para geração de endereços e 4 portas utilizadas para transmissão de dados ou controle de dispositivos, onde cada porta representa um bit.

Os componentes encoder e decoder são fabricados para oscilarem na mesma frequência. Para verificar a oscilação é necessário que ambos estejam endereçados de maneira idêntica. Para isso os pinos A1 a A5 devem estar ligados em nível lógico 1 (5 V) ou 0 (0 V), tanto no encoder quanto no decoder.

Neste trabalho foi utilizado o modo binário. Ele possibilita apenas dois estados lógicos, nível 0 (zero volts) ou nível 1 (cinco volts), tanto para as portas de endereço como para as portas de dados.

Na realização deste teste, os pinos de endereços (A1, A2, A3, A4 e A5) e de dados (D6, D7, D8 e D9) de ambos os componentes foram configurados em modo binário conforme a Tabela 3.6.

Tabela 3.6 – Endereçamento em modo binário

Pinos	Nível Lógico	
	ENCODER	DECODER
A1	0	0
A2	1	1
A3	0	0
A4	1	1
A5	0	0
D6	1	-
D7	0	-
D8	0	-
D9	0	-

Cabe observar que os pinos de dados do decoder (D6, D7, D8 e D9) estão em aberto, pois ainda não receberam a informação a ser enviada pelo encoder.

Os circuitos integrados MC 145026 e MC 145027 transmitem tanto as informações de endereço quanto as informações de dados pela transmissão via cabo.

Observa-se na Figura 3.15 que os pinos de endereços (A1, A2, A3, A4, e A5), tanto do encoder quanto do decoder, foram configurados de maneira idêntica para o teste de oscilação, com a sequência de bits 01010, sendo utilizada uma ligação direta (cabo) para efetuar a tráfego de bits entre os componentes.

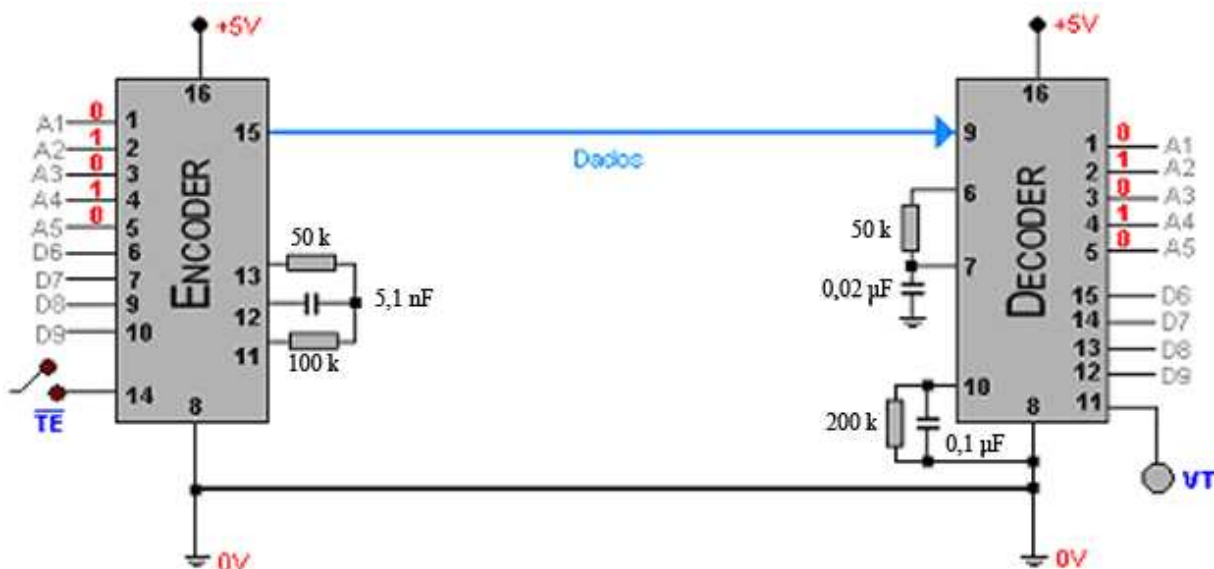


Figura 3.15 – Endereços encoder e decoder.

Para que essa transmissão ocorra, o pino 14 (TE – Transmit Enable) do encoder é levado a nível lógico 0. Com isso, os bits são enviados para o decoder, pela transmissão via cabo, contendo as informações de endereço (pinos A1 até A5) e de dados (pinos D6 até D9). A Figura 3.16 ilustra esta situação.

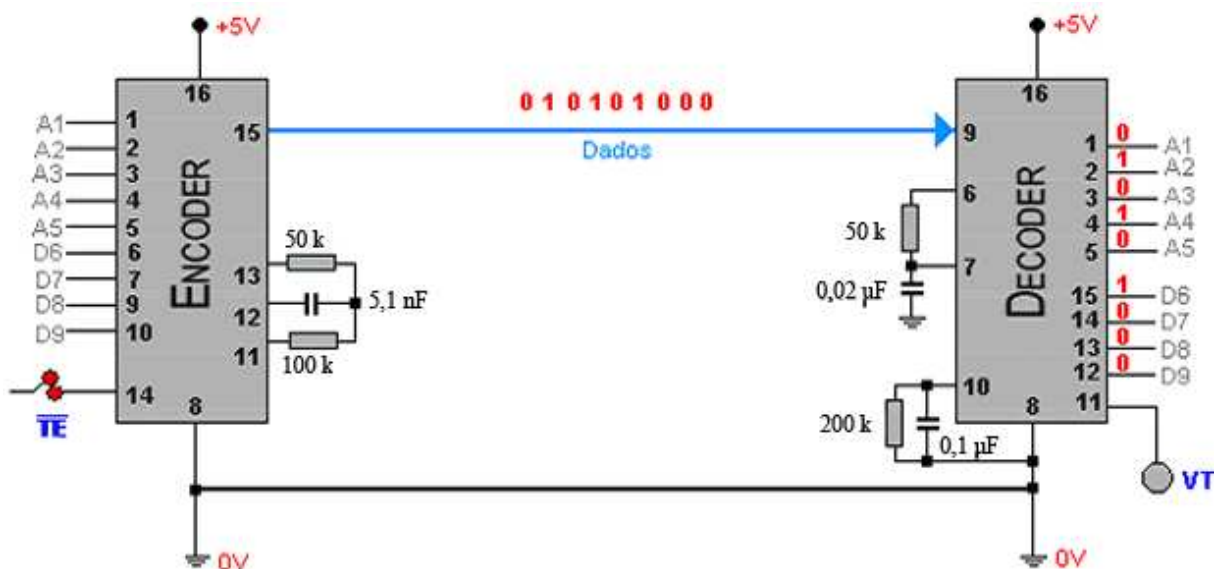


Figura 3.16 – Transmissão via cabo de dados.

Pode-se observar que os bits de endereços (01010) correspondentes aos pinos A1, A2, A3, A4 e A5 foram transmitidos juntamente com os bits de dados

(1000) referentes aos pinos D6, D7, D8 e D9. O decoder compara os bits de endereços recebidos pelo encoder com os endereços que possui. Ao fim desta comparação, se os bits de endereços forem idênticos, o decoder disponibiliza os bits de dados nos pinos de dados. Caso os bits de endereços do encoder e do decoder estejam distintos, a transmissão dos dados ocorre, sendo os bits de dados desconsiderados, não acendendo o led existente na saída do pino 11 (VT – Valid Transmission) do decoder.

O led conectado ao pino 11 do decoder (VT – Valid Transmission) deve acender para indicar que a transmissão ocorreu de forma correta. Com isso, os bits de dados são disponibilizados nos pinos D6, D7, D8 e D9. Observa-se na Figura 3.17 que o led do decoder está em amarelo, simulando o correto funcionamento e validando a transmissão.

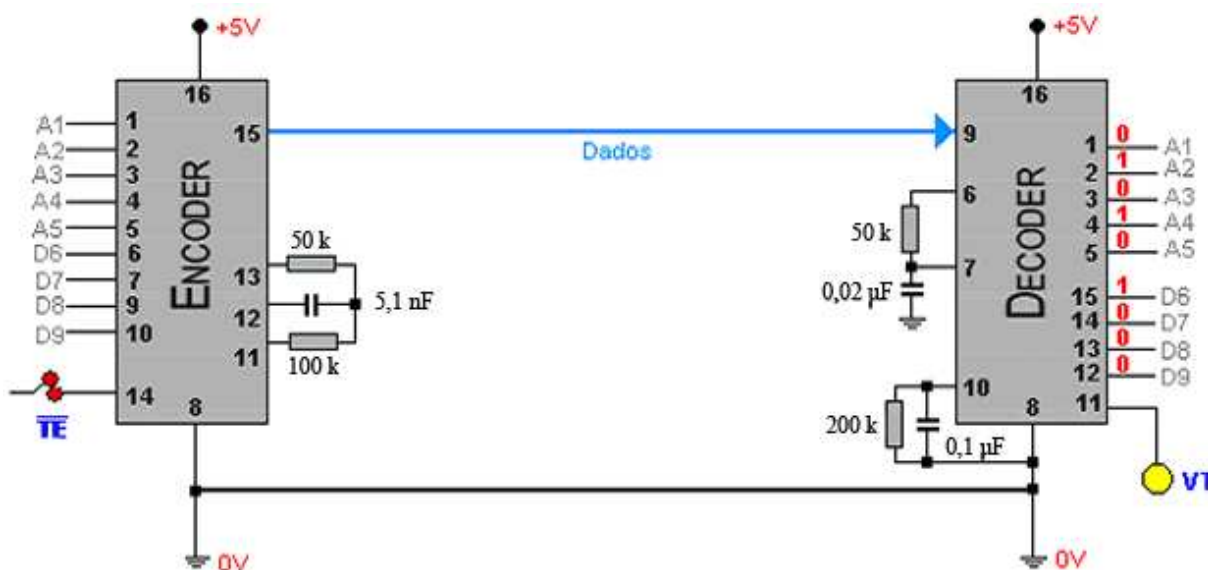


Figura 3.17 – Validação da transmissão via cabo.

Após executados os testes constatou-se o perfeito funcionamento dos componentes, utilizando o cabo de dados como meio da transmissão.

3.2.2 Teste de oscilação via Radiofrequência

Nesta etapa é feita a substituição do meio de comunicação via cabo por módulos de radiofrequência.

Para este projeto são utilizados dois conjuntos de módulos (emissor e receptor) de radiofrequência distintos: 315 MHz e 433,9 MHz, com o objetivo de inibir a interferência ocorrida entre os dois módulos utilizados quando as frequências são iguais.

Foi iniciado o teste, elevando-se o pino 14 (TE – Transmit Enable) do encoder para o nível lógico 0 (0V) com vistas a realizar a transmissão dos bits de endereços e dados do encoder para o decoder, via radiofrequência, com os novos meios de transmissão (TX e RX), conforme demonstra a Figura 3.18.

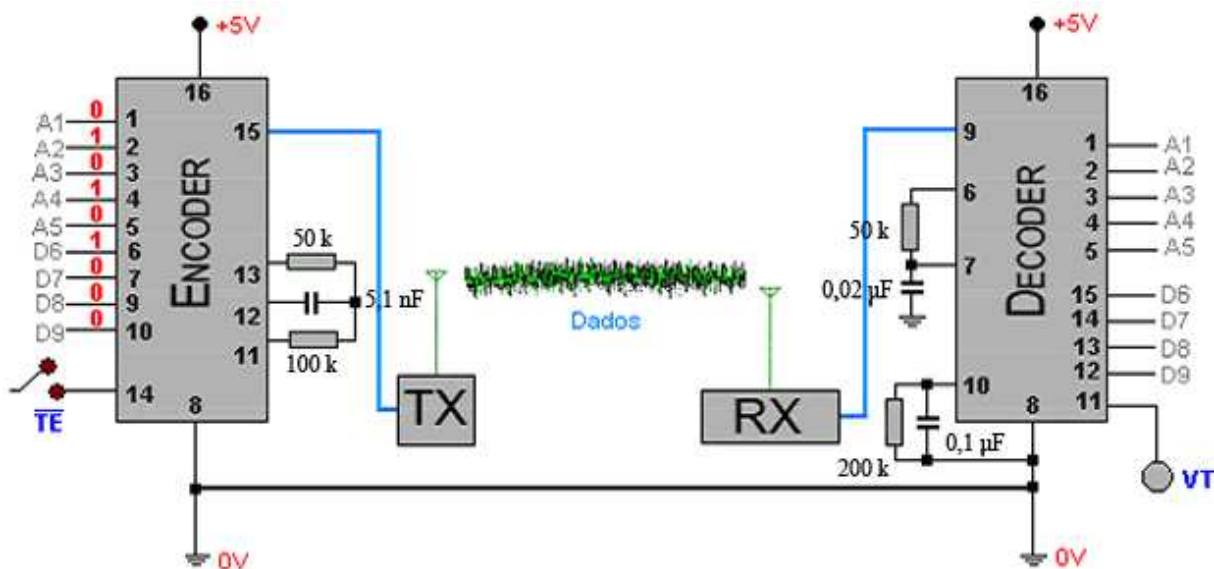


Figura 3.18 – Endereços na Transmissão RF.

Nota-se que os bits de endereços e de dados estão prontos para serem enviados para o decoder pelo módulo emissor de radiofrequência (TX).

As informações existentes nos pinos de endereços e de dados são transmitidos via radiofrequência para o decoder quando o pino 14 (TE – Transmit

Enable) do encoder for levado a nível lógico 0. Após este acionamento, o módulo transmissor (TX) transmite esses bits (endereços e dados) modulados por radiofrequência. Isto pode ser visto na Figura 3.19.

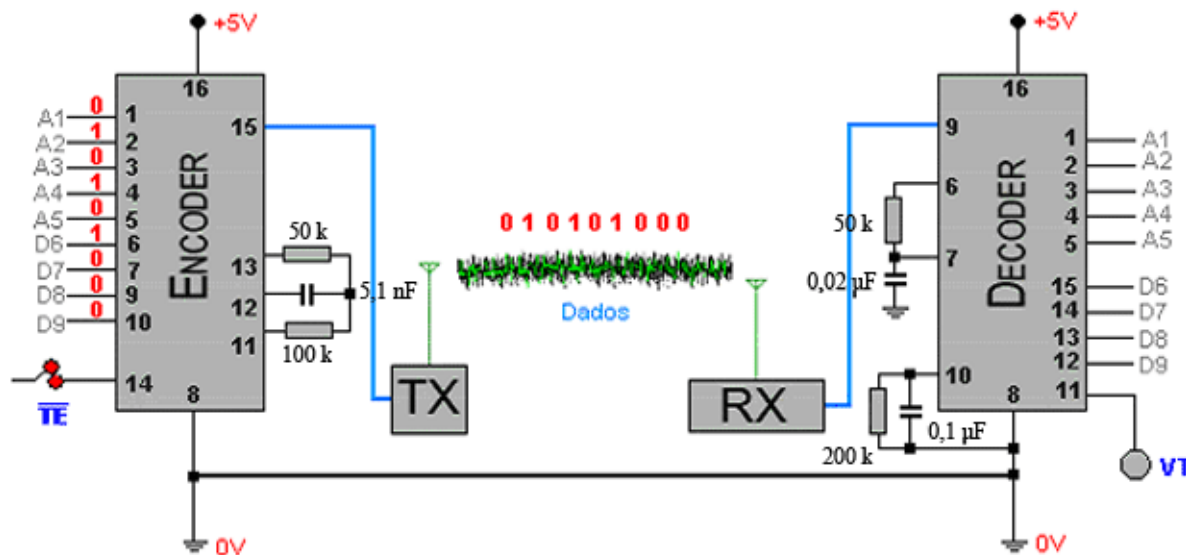


Figura 3.19 – Transmissão via RF.

O módulo receptor (RX) captura esses bits modulados e os entrega para que o decoder faça a comparação entre os bits de endereços recebidos e os bits configurados (encoder e decoder). Caso esses endereços sejam distintos dos endereços do encoder, os bits de dados são desconsiderados e o led não acende. Como os endereços estão corretos, os bits de dados são considerados e o led conectado na saída do pino 11 do decoder, que está em amarelo, simula que a transmissão ocorreu de forma correta, conforme demonstra a Figura 3.20.

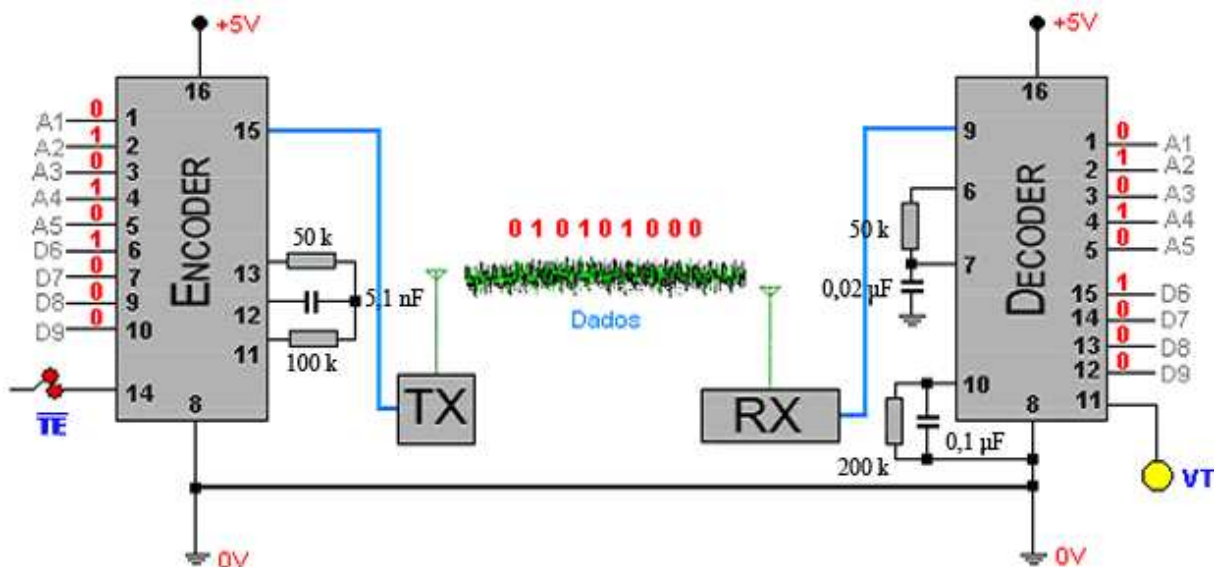


Figura 3.20 – Validação da transmissão via RF.

Assim, de forma resumida, pode-se dizer que o encoder envia os bits de endereços e de dados para o módulo transmissor no momento que o pino 14 TE (Transmission Enable) é levado a nível lógico 0 (zero volts).

O módulo emissor TX-C1 transmite os bits modulados através de radiofrequência. O módulo receptor RX-D1 captura esses dados e os repassa para o decoder para que seja feita a comparação dos bits de endereços recebidos com os endereços configurados. Caso os endereços estejam iguais, os bits de dados ficam disponíveis nos pinos de dados (D6 até D9) e o pino VT do decoder é levado a nível lógico 1 (5 volts), acendendo o led e informando que os dados foram recebidos com sucesso.

Os pinos de dados permanecem com a última informação, pois estão ligados no decoder a um tipo de memória volátil conhecido como *latch*. Esses dados permanecem nessa memória até que um novo dado seja enviado e aceito ou a até que a alimentação elétrica seja interrompida.

Na transmissão e recepção de radiofrequência o comprimento da antena é importante para que se possa aproveitar ao máximo o alcance entre os módulos transmissor e receptor.

Neste trabalho, o comprimento da antena foi de 17,28 centímetros para o módulo de 433,9 MHz, e de 23,80 centímetros para o módulo de 315 MHz, definidos de acordo a utilização da fórmula 3.1.

$$\text{Comprimento} = \frac{7500}{\text{Freq.MHz}} \quad (3.1)$$

Executados os testes, constatou-se o correto funcionamento do hardware de controle, utilizando os módulos de radiofrequência como meio de transmissão.

3.2.3 Integração dos componentes

Nesta última etapa são adicionados ao projeto o solenóide, o sensor de umidade e a comunicação via porta paralela entre o computador e o hardware desenvolvido, utilizando os módulos de radiofrequência como meio de transmissão entre o encoder e decoder.

O solenóide possui um núcleo de ferro que se move quando a corrente elétrica flui através da bobina. Quando essa corrente flui, um campo eletromagnético é gerado e move uma barra interna até o centro do solenóide, abrindo o caminho e permitindo a passagem do fluido. Quando a corrente é interrompida, uma mola empurra a barra de volta fazendo com que o caminho feche e obstrua a passagem do fluido. Este procedimento é conhecido como *push pull* e é bastante utilizado para controlar o acionamento de dispositivos.

O solenóide está integrado ao projeto conforme esquema elétrico demonstrado na Figura 3.21.

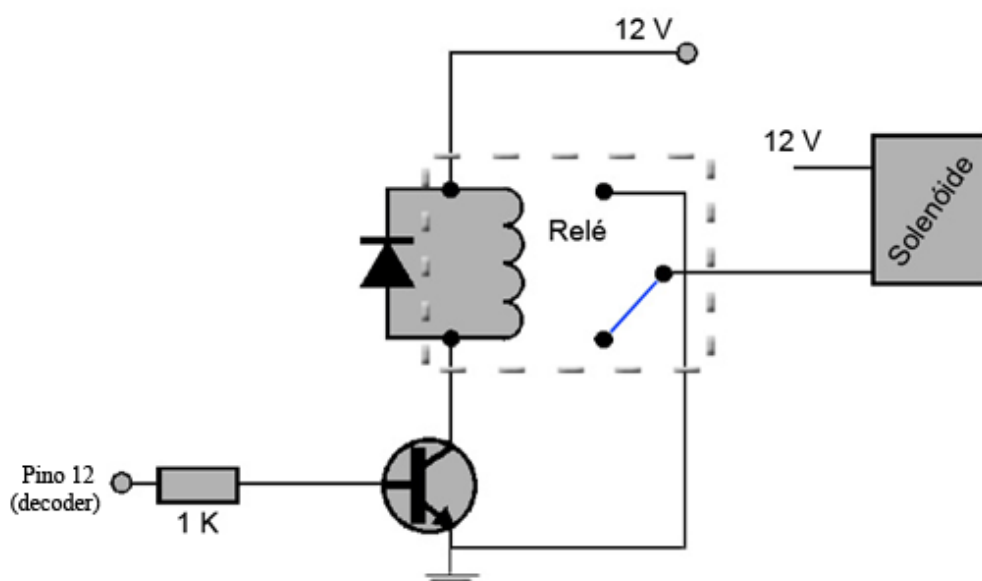


Figura 3.21 – Esquema elétrico do solenóide.

Este circuito foi ligado ao pino 12 do decoder. A função principal desse circuito é permitir ou interromper a passagem do fluxo de água para o sistema de irrigação.

Em seguida foi adicionado ao projeto o sensor de umidade, que tem a função de medir a umidade do solo através da corrente elétrica passante no meio. A tensão produzida por essa corrente é recebida pelo conversor A/D (analógico/digital) e enviada ao computador.

O computador recebe esse valor no formato digital e entrega para o software, onde serão feitas a leitura e a interpretação dos valores. Quando essa umidade medida estiver fora dos parâmetros configurados no software, o sensor solicitará o acionamento ou desligamento do sistema de irrigação.

A seguir foi conectado ao hardware o circuito integrado 74LS244, que é o buffer utilizado para a comunicação via porta paralela entre o computador e o hardware emissor de controle.

O hardware conectado à porta paralela do computador faz o processamento das informações e o acionamento do sistema de irrigação por intermédio de software

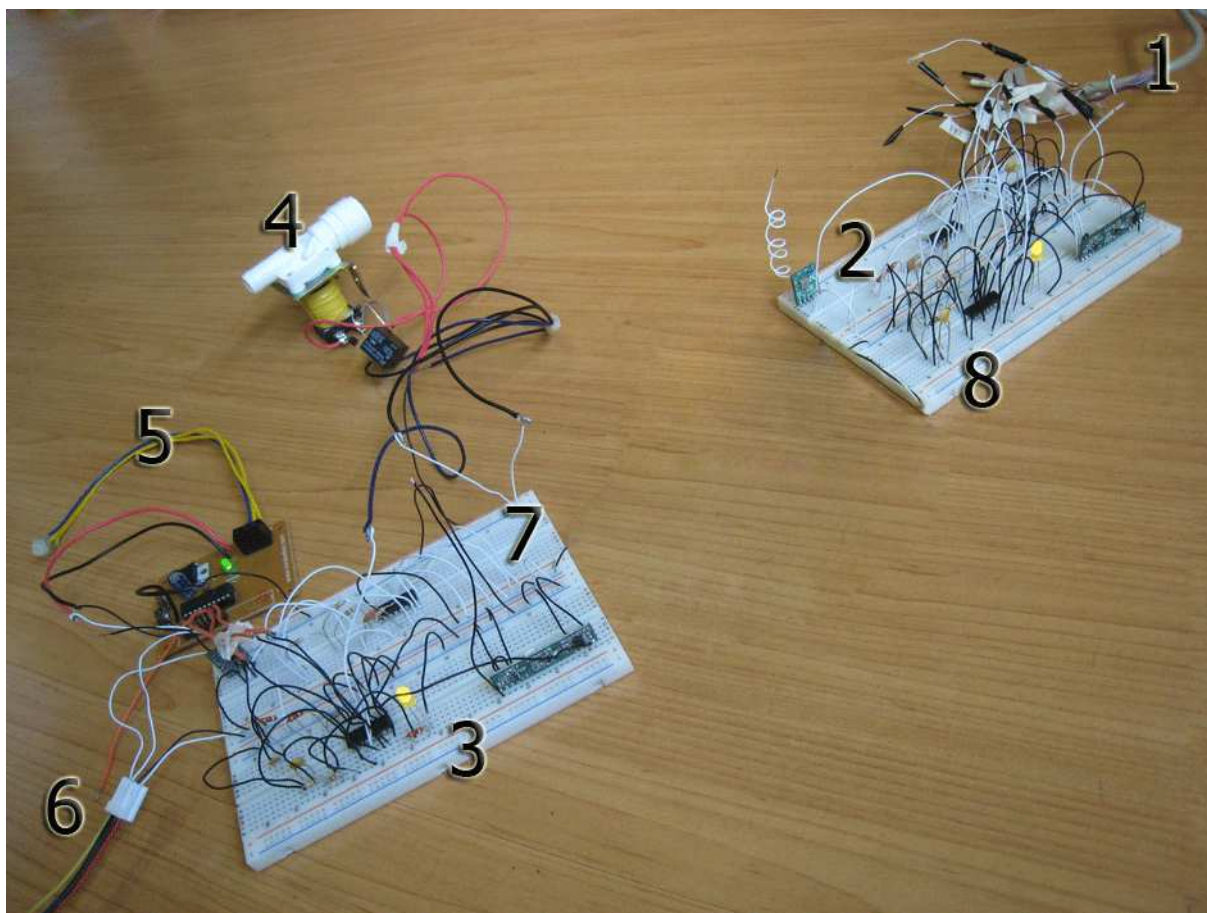


Figura 3.24 – Hardware completo.

A Tabela 3.7 descreve os componentes enumerados na Figura 4.9

Tabela 3.7 – Descrição dos itens da Figura 4.9

Item	Descrição
1	Comunicação entre o computador e o hardware desenvolvido via porta paralela.
2	Hardware para envio da solicitação de ativação/desativação do sistema de irrigação.
3	Hardware para recebimento da solicitação de ativação/desativação do sistema de irrigação.
4	Solenóide ligado ao hardware de recebimento.
5	Sensor de umidade ligado ao hardware de recebimento.
6	Cabos de alimentação de 5v e 12v dos <i>protoboards</i> .
7	Hardware para devolução do status do sistema de irrigação (ligado ou desligado) e das informações referentes ao sensor de umidade medida para interpretação do software no computador.
8	Hardware de recebimento das informações de status do sistema de irrigação e das informações referentes ao sensor de umidade.

No desenvolvimento do projeto, as definições técnicas dos circuitos integrados foram embasadas nas informações constantes nos documentos (*datasheets*) do Anexo I.

3.3 SOFTWARE

Neste tópico são descritos o funcionamento do software, bem como a tecnologia utilizada para o seu desenvolvimento.

3.3.1 Tecnologia Utilizada

Para este trabalho, são utilizados o produto Microsoft Visual Studio 2005 Professional Edition como plataforma de desenvolvimento e a linguagem de programação C# (C sharp).

3.3.2 Fluxograma

O fluxograma é uma representação simbólica que descreve uma seqüência das fases e ações do software. Tem como objetivo detalhar as etapas dos processos e permitir uma melhor visualização do software como um todo. O fluxograma do software desenvolvido nesse projeto encontra-se na Figura 3.25.

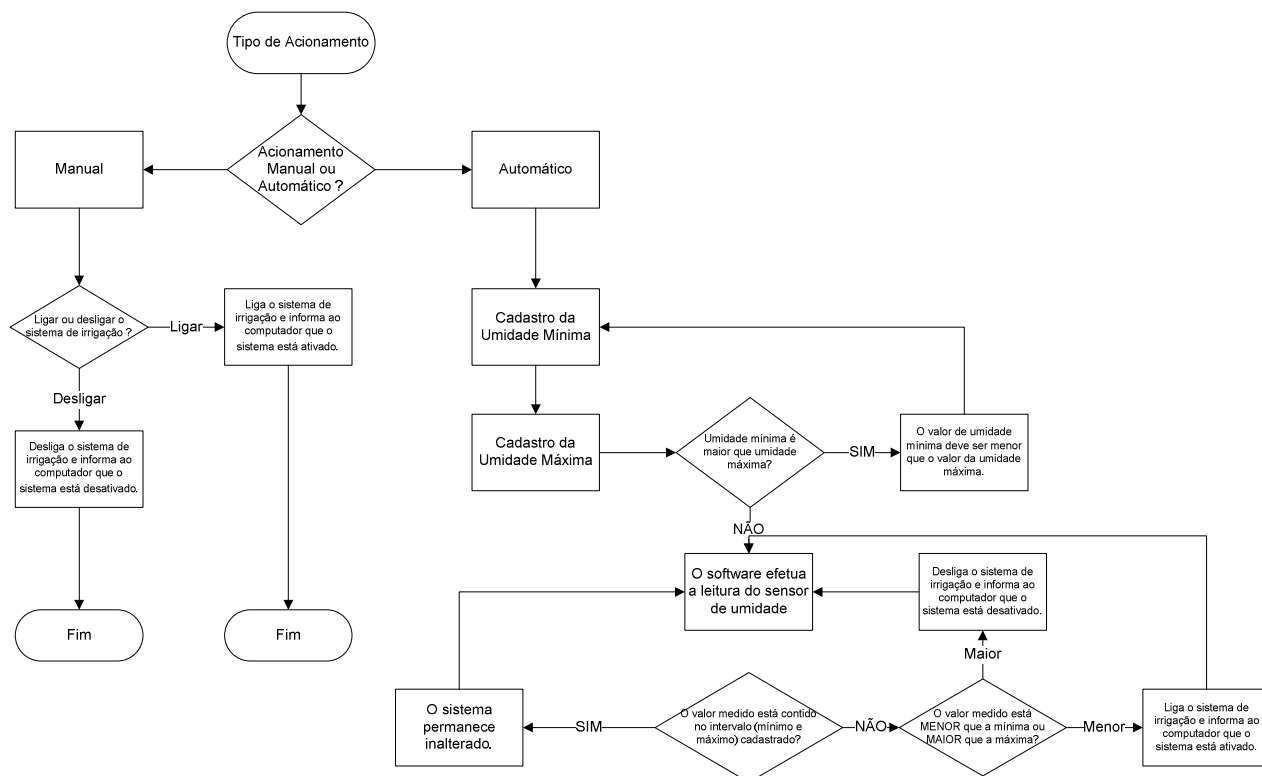


Figura 3.25 – Fluxograma do software

3.3.3 Funcionamento

O software desenvolvido é composto por várias rotinas. Todas essas rotinas são compiladas para gerarem o software final. As rotinas desenvolvidas para esse software compreendem: a rotina do módulo manual e o módulo automático, a rotina de escrita e leitura na porta paralela e a rotina de ligar e desligar o sistema de irrigação.

Durante a rotina do módulo manual, a qualquer momento, com uma ação humana pode-se solicitar o acionamento ou desligamento do sistema de irrigação. Para isto, basta clicar no botão correspondente a ação de acionar/desligar para que se execute o comando solicitado.

Para a rotina do módulo automático, o software efetua continuamente a leitura do sensor de umidade e compara o valor detectado com as umidades ideais

definidas (mínima e máxima), procedendo a ligação ou o desligamento do sistema de irrigação.

Quando o valor detectado é menor que a faixa de valores ideais de umidade definida, o software envia uma solicitação para o hardware, via porta paralela do computador, para o acionamento do sistema. Neste momento é exibida uma mensagem no computador informando que o sistema está ligado.

No caso do valor ser maior que a faixa definida, da mesma forma, é enviada uma solicitação para a interrupção do fornecimento de água para a irrigação, sendo exibida no computador uma mensagem informando que o sistema se encontra desligado.

A rotina de escrita na porta paralela é utilizada para se escrever um byte no endereço da porta paralela. Já a rotina de leitura na porta paralela serve para o software obter a informação do status do sistema de irrigação (Sistema Ligado ou Sistema Desligado).

A rotina para ligar o sistema de irrigação permite a escrita de uma seqüência de bytes na porta paralela. Da mesma forma, a rotina de desligar o sistema escreve uma seqüência de bytes, porém utilizando uma combinação de zeros (nível lógico 0) para o desligamento do sistema.

Detalhes técnicos a respeito do código fonte do software desenvolvido estão descritos no Anexo II.

4 RESULTADOS OBTIDOS

Neste capítulo são descritos os resultados obtidos nos testes efetuados no modelo desenvolvido, tanto no acionamento manual quanto no automático.

4.1 Acionamento manual do sistema

O acionamento manual do sistema obteve o resultado esperado, funcionando adequadamente, conforme ilustrado nas Figuras 4.1 a 4.4.

A Figura 4.1 representa o acionamento do sistema por intermédio do botão “LIGAR”.



Figura 4.1 – Acionamento manual.

Feito isso, o hardware emissor transmite o sinal para o módulo receptor, por meio de radiofrequência, fazendo com que os *leds* acendam, indicando a realização da transmissão e o acionamento do solenóide, conforme ilustrado na Figura 4.2.

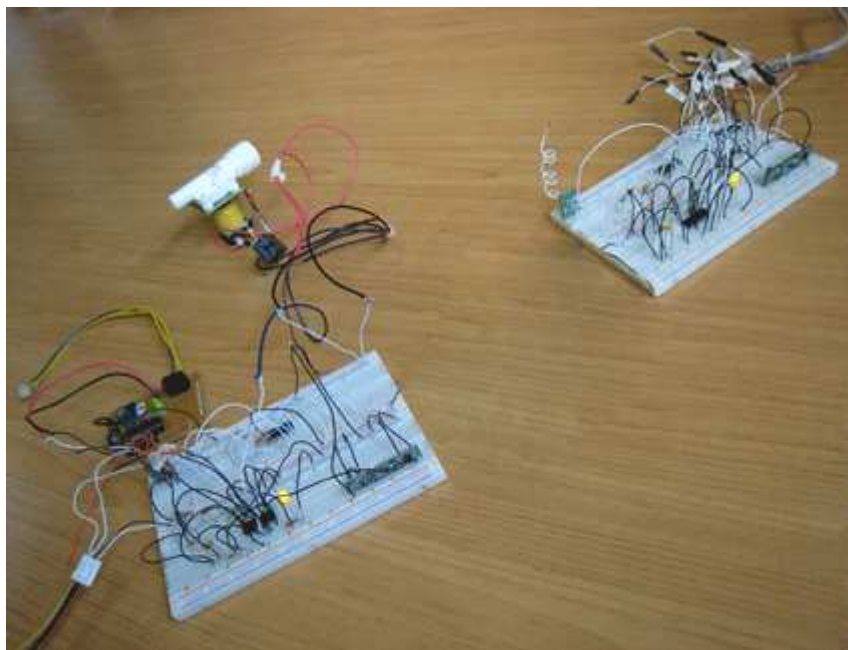


Figura 4.2 – Leds acessos e solenóide aberto.

Já a Figura 4.3 representa o desligamento manual do sistema por intermédio do botão “DESLIGAR”.



Figura 4.3 – Desligamento via software.

Dessa forma, o hardware emissor transmite o sinal para o módulo receptor, por meio de radiofrequência, fazendo com que os *leds* apaguem, indicando a realização da transmissão e a desativação do solenóide, conforme ilustrado na Figura 4.4.

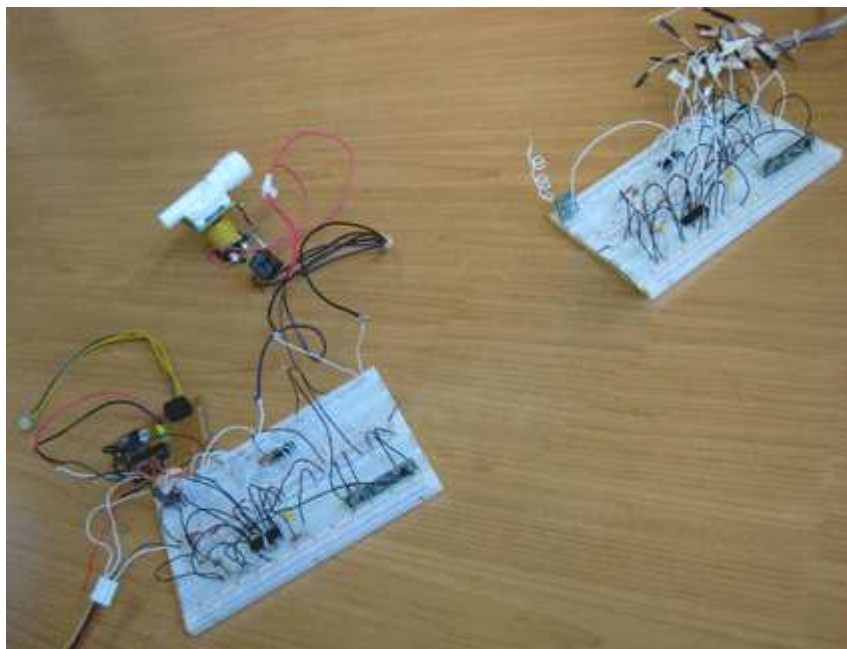


Figura 4.4 – Leds desligados e solenóide fechado.

4.2 Acionamento Automático do Sistema

O acionamento automático do sistema obteve o resultado esperado, funcionando adequadamente, conforme ilustrado nas Figuras 4.5 a 4.9.

A Figura 4.5 mostra o primeiro valor medido pelo sensor de umidade no teste efetuado.

Form1

Controle

☐ Manual ☒ Automático

Sistema Desligado

Defina a faixa de umidade (0 a 7)

3 a 5

Leitura

4

100

Daniel dos Santos Ferreira Neto
Prof. M.C Claudio Penedo

Figura 4.5 – Valor da umidade detectado pelo sensor: primeira medição.

O software efetuou a leitura do valor detectado no sensor (4) e a comparou com os valores configurados como umidades ideais, definidas como: mínima de 3 e máxima de 5. Como o valor estava dentro da faixa ideal, o sistema não foi ativado e os *leds* permaneceram apagados.

A Figura 4.6 ilustra o valor obtido em nova medição de umidade efetuada.

Form1

Controle

☐ Manual ☒ Automático

Sistema Ligado

Defina a faixa de umidade (0 a 7)

3 a 5

Leitura

2

010

Daniel dos Santos Ferreira Neto
Prof. M.C Claudio Penedo

Figura 4.6 – Valor da umidade detectado pelo sensor: segunda medição.

O software efetuou a leitura do valor detectado no sensor (2) e a comparou com os valores configurados como umidades ideais, definidas como: mínima de 3 e máxima de 5. Como o valor estava abaixo da faixa ideal, o sistema foi ativado. Foi enviado um comando de acionamento do sistema pelo hardware emissor que transmitiu esse sinal para o módulo receptor, por meio de radiofrequência, fazendo com que os *leds* fossem acessos, indicando a realização da transmissão e o acionamento do solenóide, conforme ilustrado na Figura 4.7.

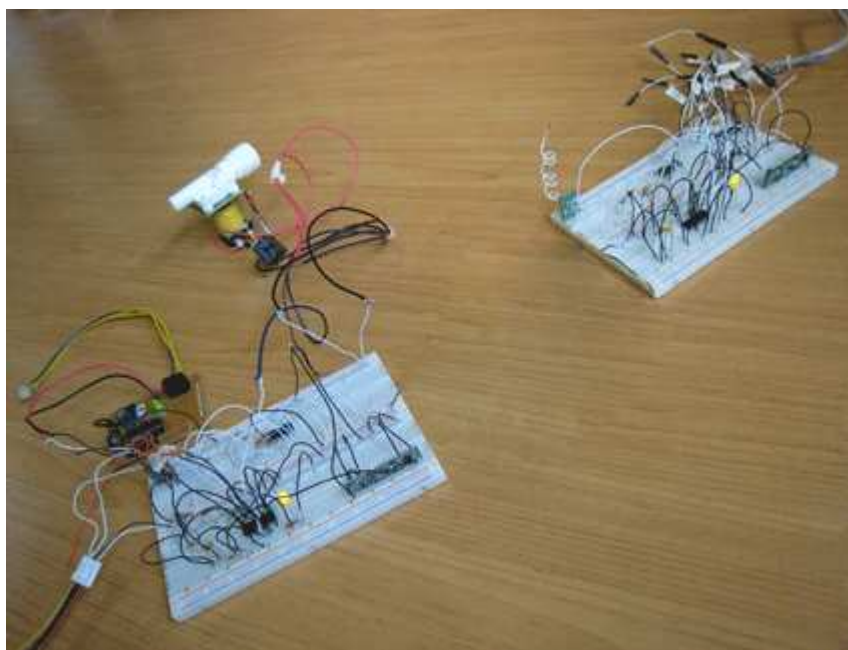


Figura 4.7 – Acionamento do sistema pelo sensor.

A seguir, nova medição do nível de umidade foi efetuada, obtendo-se o resultado mostrado na Figura 4.8.

Form1

Controle

☐ Manual ☒ Automático

Sistema Desligado

Defina a faixa de umidade (0 a 7)

3 a 5

Leitura

7

111

Daniel dos Santos Ferreira Neto
Prof. M.C Claudio Penedo

Figura 4.8 – Valor da umidade detectado pelo sensor: terceira medição.

O software efetuou a leitura do valor detectado no sensor (7) e a comparou com os valores configurados como umidades ideais, definidas como: mínima de 3 e máxima de 5. Como o valor estava acima da faixa ideal, o sistema foi desativado. Foi, então, enviado um comando para desativação do sistema pelo hardware emissor que transmitiu esse sinal para o módulo receptor, por meio de radiofrequência, fazendo com que os *leds* fossem apagados, indicando a realização da transmissão e a desativação do solenóide, conforme ilustrado na Figura 4.9.

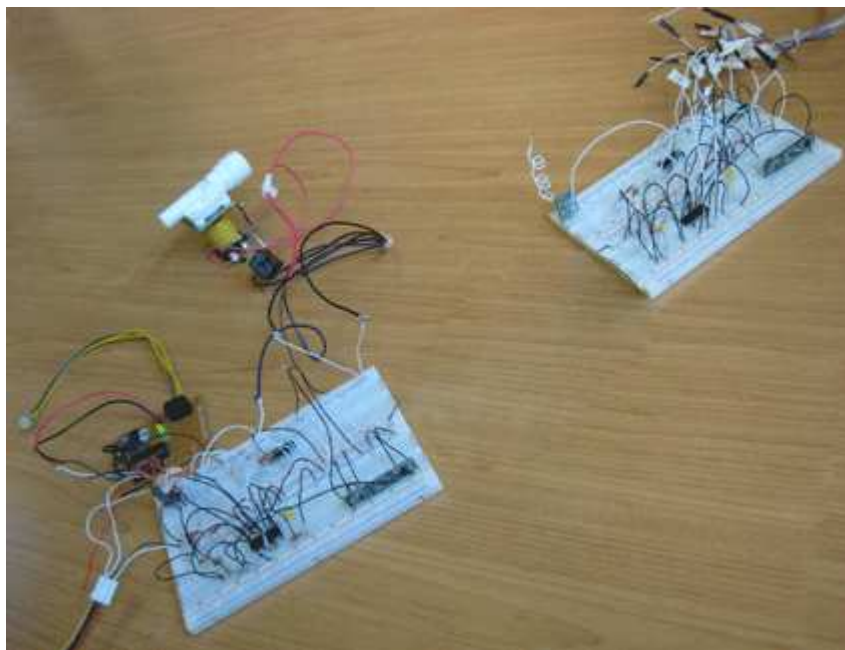


Figura 4.9 – Desativação do sistema pelo sensor.

4.3 Dificuldades Encontradas

Uma dificuldade encontrada que merece destaque diz respeito aos módulos de radiofrequência (emissor e receptor). No início do projeto foram utilizados dois módulos de mesma frequência, fato que tornava o sistema instável, apresentando interferências na comunicação entre os módulos emissores e receptores. O problema foi resolvido com a substituição de um dos módulos, de forma que eles apresentassem frequências distintas, eliminando as interferências na comunicação e a instabilidade do sistema.

Além disso, a utilização de *protoboards* para elaboração dos circuitos eletrônicos mostrou-se suscetível a maus contatos na conexão dos fios a placa, ocasionando várias falhas durante as fases de desenvolvimento e testes do projeto, que tinham de ser identificadas e corrigidas a cada ocorrência de defeito.

5 CONCLUSÃO

Neste trabalho foi desenvolvido e testado um protótipo acadêmico, com a implementação de hardware e software, com o propósito de automatizar os procedimentos de acionamento do dispositivo que controla a passagem/interrupção do fluxo da água para o sistema de irrigação.

Na implementação do modelo foram utilizados um hardware para emissão e outro para o recebimento do sinal via radiofrequência, acionado por software, de forma manual ou automática, quando solicitado pelo sensor de umidade, via porta paralela do computador.

O sensor efetuou, de forma contínua, a leitura da umidade do solo. Os dados obtidos na leitura foram transmitidos, via radiofrequência, do hardware emissor para o receptor, que está conectado ao computador pela porta paralela. Então, o software desenvolvido comparou os valores recebidos com os índices de umidade ideais previamente definidos, emitindo o comando para acionamento do dispositivo que permite a passagem/interrupção da água.

Os resultados obtidos nos testes efetuados mostraram o funcionamento adequado do protótipo desenvolvido, de acordo com os objetivos propostos por este trabalho. Como esperado, o modelo obtido demonstrou ser um sistema simples, que permite acionar e desligar, de forma automática e programada, o fornecimento de água no processo de irrigação.

Portanto, constatou-se ser um modelo eficiente, prático e de fácil utilização, que possibilita um melhor aproveitamento da água na irrigação, além de reduzir a atuação humana por meio da automatização do processo.

Além da introdução e da conclusão, o trabalho foi estruturado da seguinte forma: no capítulo 2 foram apresentados os conceitos teóricos sobre as diversas técnicas, sistemas e métodos de irrigação existentes; por sua vez, o capítulo 3 descreveu os componentes eletrônicos utilizados no projeto, os processos de

construção dos hardwares e do software e o funcionamento do modelo proposto; por fim, o capítulo 4 apresentou os resultados obtidos neste projeto.

Os custos para a implementação do projeto, considerados os gastos relativos aos equipamentos e componentes de hardware envolvidos, e o desenvolvimento do software utilizado são estimados em aproximadamente R\$ 3.000,00 (três mil reais).

Finalmente, registra-se a pretensão futura de: (i) substituir os dispositivos encoder/decoder por circuitos de oito bits, que possibilitará a obtenção de maior precisão na medição da umidade, (ii) adicionar mais módulos de recebimento (decoder), com vistas a controlar mais sistemas de irrigação, aumentando assim a área de abrangência de utilização do projeto, (iii) implementar o projeto em placa de circuito impresso e utilizando microcontrolador e (iv) adicionar sensor de umidade do ar colocando em vários pontos.

REFERÊNCIAS

BARRETO, Geraldo Benedito. **“Irrigação: princípios, métodos e pratica”**. Campinas, Instituto Campineiro de Ensino Agrícola: Editora Arte Gráfica, São Paulo, 1974.

JACOB, M., HALKIAS, C. **“Eletronic devices and circuits - International Student Edition”**. Editora LITEC – Livraria Técnica Ltda, 1967.

JACOB, M., HALKIAS, C. **“Eletrônica: dispositivos e Circuitos”**. 4ª Edição. São Paulo: Editora McGraw-Hill do Brasil.

OLITTA, Antonio Fernando Lordelo. **“Os métodos de irrigação”**. São Paulo: Editora Nobel, 1977.

TEXAS INSTRUMENTS INCORPORATED. **“Ttl Logic Data Book: Standard Ttl, Schottky, Low-Power Schottky Circuits”**. EUA: Editora LITEC – Livraria Técnica Ltda, 1988.

VIEIRA, Dirceu Brasil. **“As técnicas de irrigação”**. São Paulo: Editora Globo, 1989.

WATERS, F. J., VALENTE, R. B. **“ABC dos componentes eletrônicos”**. São Paulo: Editora Antenna, 1979.

ANEXO I – DATASHEET DOS COMPONENTES UTILIZADOS NO HARDWARE

MOTOROLA SEMICONDUCTOR TECHNICAL DATA

Order this document
by MC145026/D

Encoder and Decoder Pairs CMOS

These devices are designed to be used as encoder/decoder pairs in remote control applications.

The MC145026 encodes nine lines of information and serially sends this information upon receipt of a transmit enable (TE) signal. The nine lines may be encoded with trinary data (low, high, or open) or binary data (low or high). The words are transmitted twice per encoding sequence to increase security.

The MC145027 decoder receives the serial stream and interprets five of the trinary digits as an address code. Thus, 243 addresses are possible. If binary data is used at the encoder, 32 addresses are possible. The remaining serial information is interpreted as four bits of binary data. The valid transmission (VT) output goes high on the MC145027 when two conditions are met. First, two addresses must be consecutively received (in one encoding sequence) which both match the local address. Second, the 4 bits of data must match the last valid data received. The active VT indicates that the information at the Data output pins has been updated.

The MC145028 decoder treats all nine trinary digits as an address which allows 19,683 codes. If binary data is encoded, 512 codes are possible. The VT output goes high on the MC145028 when two addresses are consecutively received (in one encoding sequence) which both match the local address.

- Operating Temperature Range: -40 to $+85^{\circ}\text{C}$
- Very-Low Standby Current for the Encoder: 300 nA Maximum @ 25°C
- Interfaces with RF, Ultrasonic, or Infrared Modulators and Demodulators
- RC Oscillator, No Crystal Required
- High External Component Tolerance; Can Use $\pm 5\%$ Components
- Internal Power-On Reset Forces All Decoder Outputs Low
- Operating Voltage Range: MC145026 = 2.5 to 18 V*

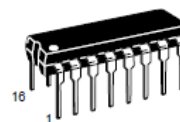
MC145027, MC145028 = 4.5 to 18 V

- For Infrared Applications, See Application Note AN1016/D

PIN ASSIGNMENTS

MC145026 ENCODER			MC145027 DECODERS			MC145028 DECODERS		
A1	1	V _{DD}	A1	1	V _{DD}	A1	1	V _{DD}
A2	2	D _{out}	A2	2	D ₆	A2	2	A ₆
A3	3	TE	A3	3	D ₇	A3	3	A ₇
A4	4	R _{TC}	A4	4	D ₈	A4	4	A ₈
A5	5	C _{TC}	A5	5	D ₉	A5	5	A ₉
A6/D6	6	R _S	R ₁	6	VT	R ₁	6	VT
A7/D7	7	A ₉ /D ₉	C ₁	7	R ₂ /C ₂	C ₁	7	R ₂ /C ₂
V _{SS}	8	A ₈ /D ₈	V _{SS}	8	D _{in}	V _{SS}	8	D _{in}
	9			9			9	

MC145026
MC145027
MC145028



P SUFFIX
PLASTIC DIP
CASE 648



D SUFFIX
SOG PACKAGE
CASE 751B



DW SUFFIX
SOG PACKAGE
CASE 751G

ORDERING INFORMATION

MC145026P	Plastic DIP
MC145026D	SOG Package
MC145027P	Plastic DIP
MC145027DW	SOG Package
MC145028P	Plastic DIP
MC145028DW	SOG Package



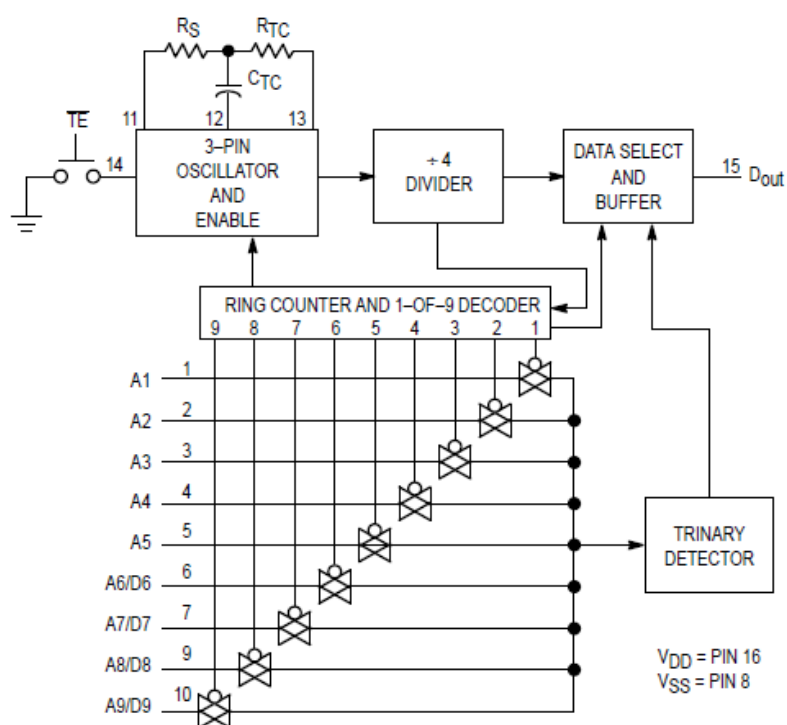


Figure 1. MC145026 Encoder Block Diagram

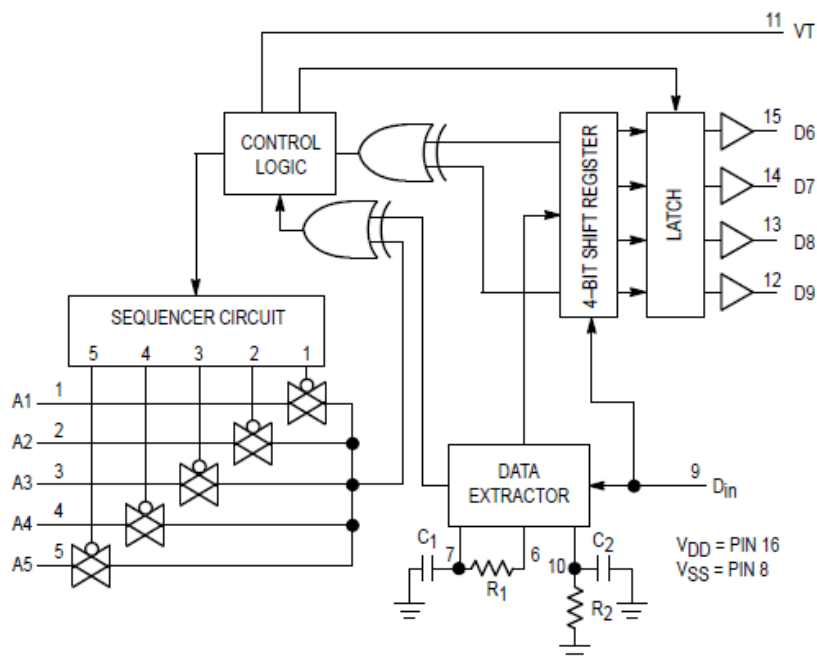


Figure 2. MC145027 Decoder Block Diagram

SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244 SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

SDLS144 - APRIL 1985 - REVISED MARCH 1988

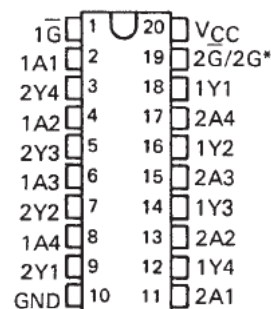
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- PNP Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

description

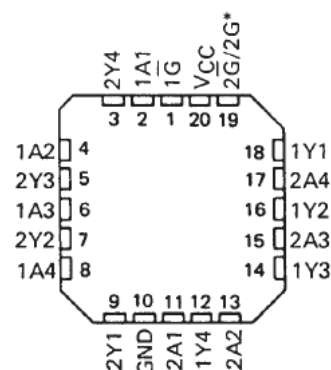
These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical \bar{G} (active-low output control) inputs, and complementary G and \bar{G} inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

The SN54' family is characterized for operation over the full military temperature range of -55°C to 125°C . The SN74' family is characterized for operation from 0°C to 70°C .

SN54LS', SN54S' . . . J OR W PACKAGE
 SN74LS', SN74S' . . . DW OR N PACKAGE
 (TOP VIEW)

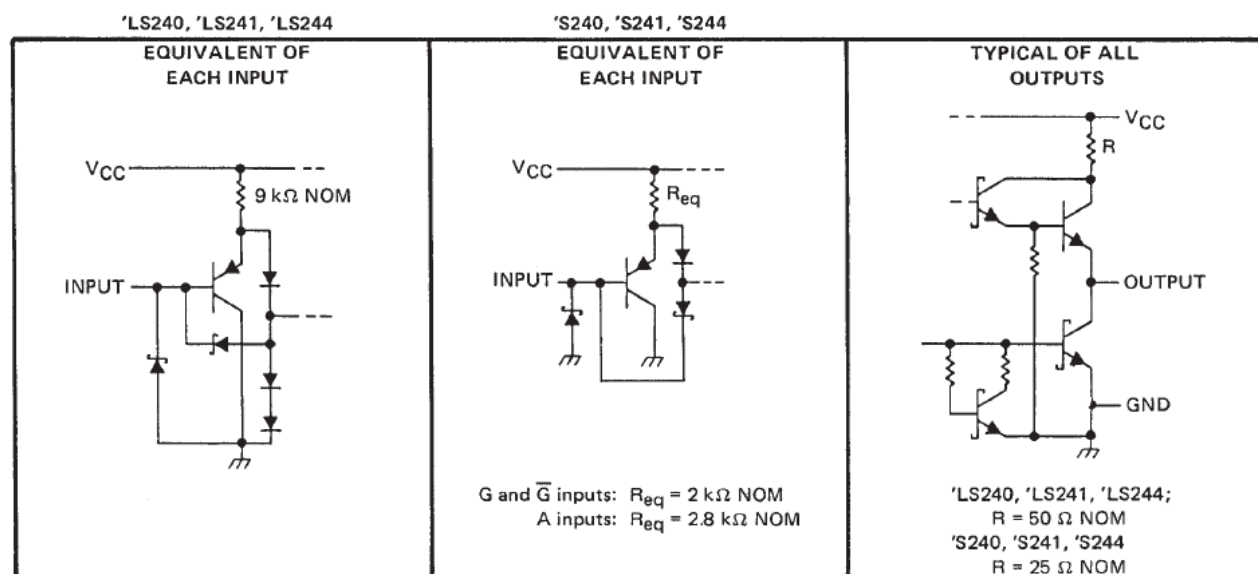


SN54LS', SN54S' . . . FK PACKAGE
 (TOP VIEW)



*2G for 'LS241 and 'S241 or 2G for all other drivers.

schematics of inputs and outputs



PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS

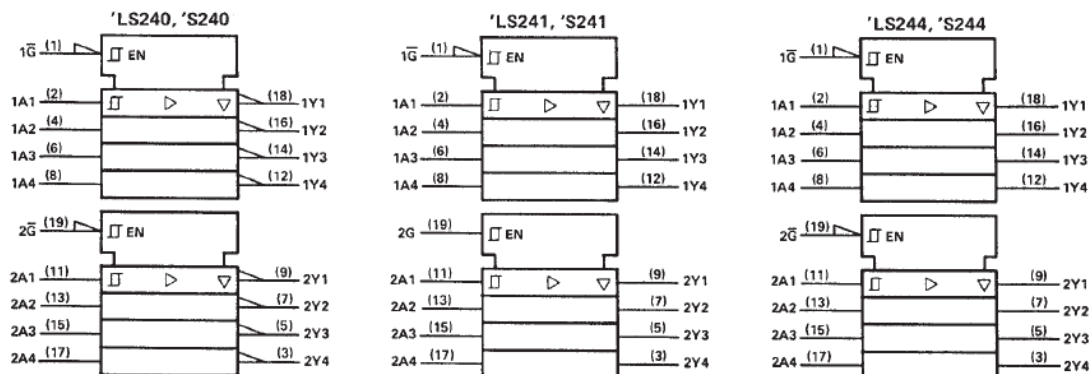
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1988, Texas Instruments Incorporated

SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244 **SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244** **OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS**

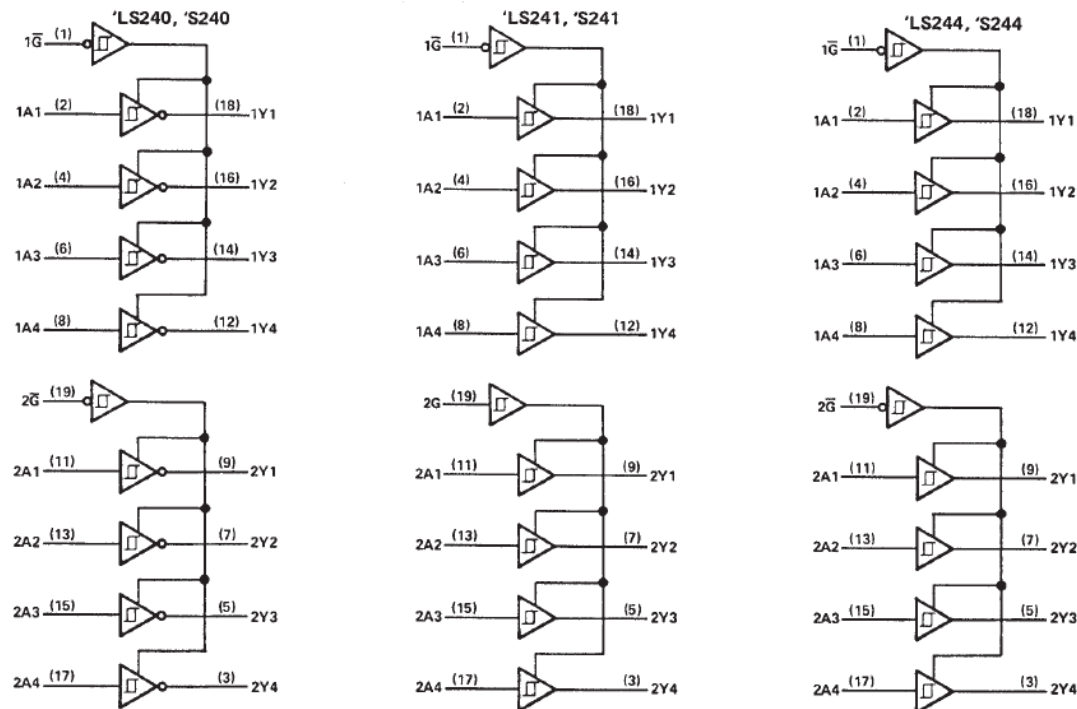
SDLS144 – APRIL 1985 – REVISED MARCH 1988

logic symbols†



†These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.

logic diagrams (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: 'LS Circuits	7 V
'S Circuits	5.5 V
Off-state output voltage	5.5 V
Operating free-air temperature range: SN54LS', SN54S' Circuits	– 55°C to 125°C
SN74LS', SN74S' Circuits	0°C to 70°C
Storage temperature range	– 65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

**SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244
SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS**

SDLS144 – APRIL 1985 – REVISED MARCH 1988

recommended operating conditions

PARAMETER	SN54LS'			SN74LS'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
V_{IH} High-level input voltage	2			2			V
V_{IL} Low-level input voltage			0.7			0.8	V
I_{OH} High-level output current			-12			-15	mA
I_{OL} Low-level output current			12			24	mA
T_A Operating free-air temperature	-55		125	0		70	°C

NOTE 1: Voltage values are with respect to network ground terminal.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†		SN54LS'			SN74LS'			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	$V_{CC} = \text{MIN}, I_I = -18 \text{ mA}$				-1.5			-1.5	V
Hysteresis ($V_{T+} - V_{T-}$)	$V_{CC} = \text{MIN}$		0.2	0.4		0.2	0.4		V
V_{OH}	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = \text{MAX}, I_{OH} = -3 \text{ mA}$		2.4	3.4		2.4	3.4		V
	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.5 \text{ V}, I_{OH} = \text{MAX}$		2			2			
V_{OL}	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = \text{MAX}, I_{OL} = 12 \text{ mA}$				0.4			0.4	V
	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = \text{MAX}, I_{OL} = 24 \text{ mA}$							0.5	
I_{OZH}	$V_{CC} = \text{MAX}, V_{IH} = 2 \text{ V}, V_{OL} = 2.7 \text{ V}$				20			20	μA
I_{OZL}	$V_{CC} = \text{MAX}, V_{IL} = \text{MAX}, V_{OL} = 0.4 \text{ V}$				-20			-20	
I_I	$V_{CC} = \text{MAX}, V_I = 7 \text{ V}$				0.1			0.1	mA
I_{IH}	$V_{CC} = \text{MAX}, V_I = 2.7 \text{ V}$				20			20	μA
I_{IL}	$V_{CC} = \text{MAX}, V_{IL} = 0.4 \text{ V}$				-0.2			-0.2	mA
$I_{OS}§$	$V_{CC} = \text{MAX}$		-40		-225	-40		-225	mA
I_{CC}	Outputs high	All		17	27		17	27	mA
	Outputs low	'LS240		26	44		26	44	
	All outputs disabled	'LS241, 'LS244		27	46		27	46	
		'LS240		29	50		29	50	
		'LS241, 'LS244		32	54		32	54	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}, T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS		'LS240			'LS241, 'LS244			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	$R_L = 667 \Omega, C_L = 45 \text{ pF},$ See Note 2			9	14		12	18	ns
t_{PHL}				12	18		12	18	ns
t_{PZL}				20	30		20	30	ns
t_{PZH}				15	23		15	23	ns
t_{PLZ}	$R_L = 667 \Omega, C_L = 5 \text{ pF},$ See Note 2			10	20		10	20	ns
t_{PHZ}				15	25		15	25	ns

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244 SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

SDLS144 – APRIL 1985 – REVISED MARCH 1988

recommended operating conditions

PARAMETER	SN54S'			SN74S'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage, (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
V_{IH} High-level input voltage	2			2			V
V_{IL} Low-level input voltage			0.8			0.8	V
I_{OH} High-level output current			-12			-15	mA
I_{OL} Low-level output current			48			64	mA
External resistance between any input and V_{CC} or ground			40			40	k Ω
T_A Operating free-air temperature (see Note 3)	-55		125	0		70	$^{\circ}$ C

NOTES: 1. Voltage values are with respect to network ground terminal.

3. An SN54S241J operating at free-air temperature above 116 $^{\circ}$ C requires a heat sink that provides a thermal resistance from case to free-air $R_{\theta CA}$ of not more than 40 $^{\circ}$ C/W.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		SN54S'		SN74S'		UNIT
				MIN	TYP‡	MAX	MIN	
V _{IK}		V _{CC} = MIN, I _I = -18 mA		-1.2		-1.2		V
Hysteresis (V _{T+} - V _{T-})		V _{CC} = MIN		0.2	0.4	0.2	0.4	V
V _{OH}		V _{CC} = MIN, I _{OH} = -1 mA	V _{IH} = 2 V, V _{IL} = 0.8 V,			2.7		V
		V _{CC} = MIN, I _{OH} = -3 mA	V _{IH} = 2 V, V _{IL} = 0.8 V,	2.4	3.4	2.4	3.4	
		V _{CC} = MIN, I _{OH} = MAX	V _{IH} = 2 V, V _{IL} = 0.5 V,	2		2		
V _{OL}		V _{CC} = MIN, I _{OL} = MAX	V _{IH} = 2 V, V _{IL} = 0.8 V,	0.55		0.55		V
I _{OZH}		V _{CC} = MAX, V _{IH} = 2 V,	V _O = 2.4 V	50		50		μA
I _{OZL}		V _{IL} = 0.8 V,	V _O = 0.5 V	-50		-50		
I _I		V _{CC} = MAX, V _I = 5.5 V		1		1		mA
I _{IH}		V _{CC} = MAX, V _I = 2.7 V		50		50		μA
I _{IL}	Any A	V _{CC} = MAX, V _I = 0.5 V		-400		-400		μA
	Any G			-2		-2		mA
I _{OS} §		V _{CC} = MAX		-50	-225	-50	-225	mA
I _{CC}	Outputs high	V _{CC} = MAX, Outputs open	'S240	80	123	80	135	mA
	'S241, 'S244		95	147	95	160		
	Outputs low		'S240	100	145	100	150	
	'S241, 'S244		120	170	120	180		
	Outputs disabled		'S240	100	145	100	150	
			'S241, 'S244	120	170	120	180	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.



Hybrid Transmitter

TX - C1

 315/433.92MHZ
Hybrid
Transmitter

- ◎Ideal for 315/433.92MHz Remote Keyless-Entry Transmitter.
- ◎SAW RESONATOR
- ◎ASK Design

The TXC1 is an ASK transmitter module .The result is excellent performance in a simple-to-use .The TXC1 is designed specifically for remote-control , wireless mouse and car alarm system operating at 315/433.92 Mhz in the USA under FCC Part 15 regulation.

Absolute Maximum Ratings

Parameter	Value	Units
Power Supply and/or Modulation Input Voltage	3	V
Operating temperature	-20 to +60	°C

Receiver Characteristics

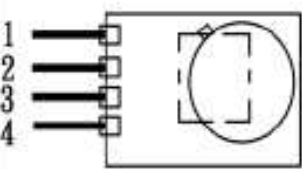
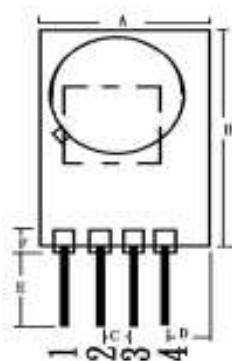
Parameter	Symbol	Condition		Value			Unit
				min.	typ.	max.	
Output power		Vcc=3.0V,TA=27°C, 50Ωload	315MHz	2	3	6	dBm
			434MHz	1	3	6	dBm
Supply current	Icc			9	10	19	mA
Supply voltage Range	Vcc				3		V
Data Rate				300	1K	10K	bps

315/434 MHz

Hybrid Transmitter

Pin assignment

PIN	Connections
1	GND
2	DATA
3	VCC
4	ANT

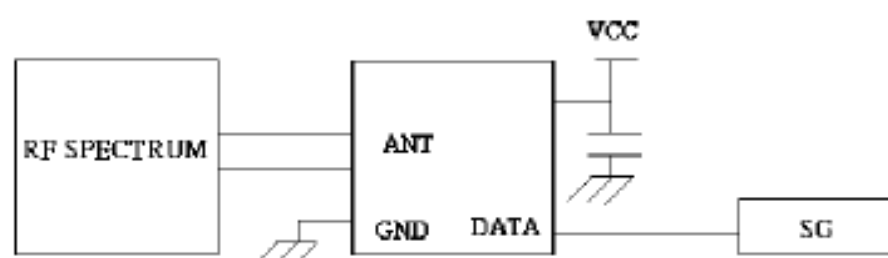



Dimensions	Millimeters	Dimensions	Millimeters
A	14.5 + 0.25mm	F	0.1 + 0.05mm
B	18.5 + 0.25mm	G	0.04 + 0.005mm
C	1.8 + 0.05mm	H	4.2 + (MAX)
D	5.0 + 0.1mm		
E	6.5 + 0.1mm		

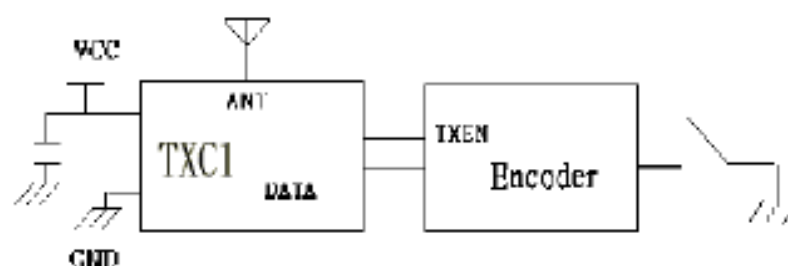
315/434 MHz

Hybrid Transmitter

Typing Test Circuit



Typical transmitter Application



Notes:

1. Encoder : HT12D/F , PTC (2262)
2. Antenna : Length = 22.6cm for 315MHz ; Length = 17.2 cm for 434MHz.



Hybrid Receiver

RXD1

315/434 MHz
Hybrid
Receiver

- Ideal for 315/434MHz Remote Keyless-Entry Receivers.

The RXD1 is a miniature receiver module that receives On-off keyed (OOK) modulation signal and demodulated to digital signal for the next decoder stage. Local Oscillator is made of LC structure. The result is excellent performance in a simple-to-use, with a low external component count. The RXD1 is designed specifically for remote-control and wireless security receiver operating at 315/434MHz in the USA under FCC Part 15 regulation.

Absolute Maximum Ratings

Rating	Value	Units
Power Supply and/or Modulation Input Voltage	5	V
Operating temperature	-20 to +80	°C

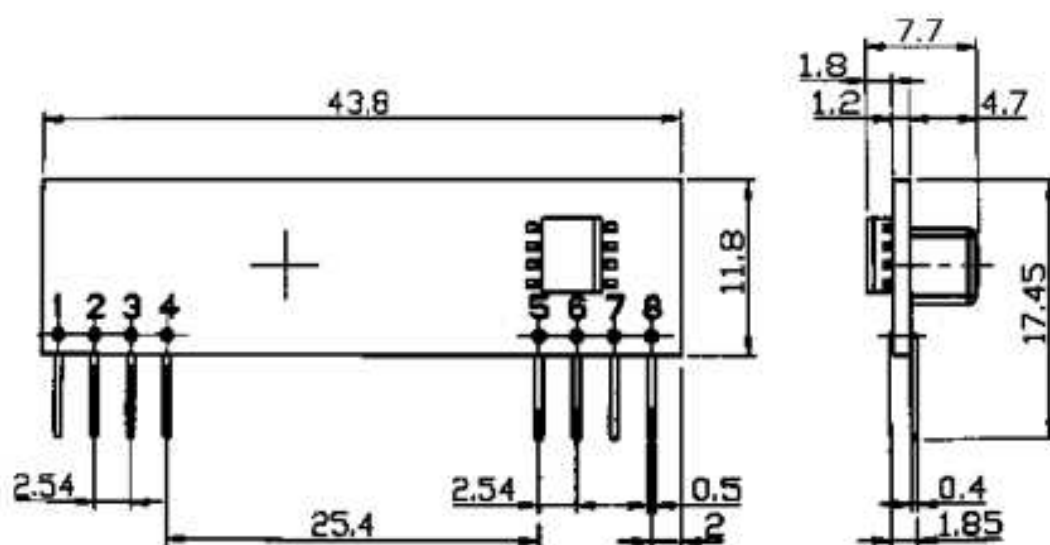
Receiver Characteristics

Parameter	Symbol	Condition	Value			Unit
			min.	typ.	max.	
Sensitivity	P _{sens}	V _{cc} =5.0V, AT25 °C, BER=3/100, 2Kbps	315MHz	-105	-103	dBm
			434MHz	-104	-102	dBm
ASKOUT logic HIGH	VOH	I _{load} = 10uA	0.7*V _{cc}			V
ASKOUT logic LOW	VOL	I _{load} = 10uA			0.3*V _{cc}	V
Supply current	I _{cc}			3.3	3.6	mA
Supply voltage Range	V _{cc}		+4.75	+5	+5.25	V
Data Rate			200	2K	6K	bps
RF Bandwidth-3dB				4M		Hz

315/434 MHz

Hybrid Receiver

Mechanical Characteristics:



Pin-out:

1.ANT	5.VCC
1.GND	6.DATA
2.GND	7.DATA
3.VCC	8.GND

ANEXO II – CÓDIGO FONTE DO SOFTWARE

PROGRAMAÇÃO DO SISTEMA

```
namespace SisIrigacao
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.btnligar = new System.Windows.Forms.Button();
            this.labell = new System.Windows.Forms.Label();
            this.txtLeitura = new System.Windows.Forms.TextBox();
            this.btndesligar = new System.Windows.Forms.Button();
            this.timer1 = new System.Windows.Forms.Timer(this.components);
```



```

this.lblStatus = new System.Windows.Forms.Label();
this.grpControle = new System.Windows.Forms.GroupBox();
this.rdManual = new System.Windows.Forms.RadioButton();
this.rdAutomatico = new System.Windows.Forms.RadioButton();
this.lblUmidade = new System.Windows.Forms.Label();
this.MinUmidade = new System.Windows.Forms.TextBox();
this.label2 = new System.Windows.Forms.Label();
this.MaxUmidade = new System.Windows.Forms.TextBox();
this.pnUmidade = new System.Windows.Forms.Panel();
this.textTeste = new System.Windows.Forms.TextBox();
this.label3 = new System.Windows.Forms.Label();
this.label4 = new System.Windows.Forms.Label();
this.grpControle.SuspendLayout();
this.pnUmidade.SuspendLayout();
this.SuspendLayout();
//
// btnligar
//
this.btnligar.Location = new System.Drawing.Point(60, 301);
this.btnligar.Name = "btnligar";
this.btnligar.Size = new System.Drawing.Size(75, 23);
this.btnligar.TabIndex = 0;
this.btnligar.Text = "LIGAR";
this.btnligar.UseVisualStyleBackColor = true;
this.btnligar.Click += new
System.EventHandler(this.button1_Click);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(406, 159);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(39, 13);
this.label1.TabIndex = 1;
this.label1.Text = "Leitura";
//
// txtLeitura
//
this.txtLeitura.Location = new System.Drawing.Point(409, 175);
this.txtLeitura.Name = "txtLeitura";
this.txtLeitura.Size = new System.Drawing.Size(117, 20);

```

```

this.txtLeitura.TabIndex = 4;
//
// btndesligar
//
this.btndesligar.Location = new System.Drawing.Point(243, 301);
this.btndesligar.Name = "btndesligar";
this.btndesligar.Size = new System.Drawing.Size(75, 23);
this.btndesligar.TabIndex = 5;
this.btndesligar.Text = "DESLIGAR";
this.btndesligar.UseVisualStyleBackColor = true;
this.btndesligar.Click += new
System.EventHandler(this.button2_Click);
//
// timer1
//
this.timer1.Enabled = true;
this.timer1.Interval = 500;
this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
//
// lblStatus
//
this.lblStatus.AutoSize = true;
this.lblStatus.Location = new System.Drawing.Point(413, 24);
this.lblStatus.Name = "lblStatus";
this.lblStatus.Size = new System.Drawing.Size(37, 13);
this.lblStatus.TabIndex = 7;
this.lblStatus.Text = "Status";
//
// grpControle
//
this.grpControle.Controls.Add(this.rdManual);
this.grpControle.Controls.Add(this.rdAutomatico);
this.grpControle.Location = new System.Drawing.Point(12, 24);
this.grpControle.Name = "grpControle";
this.grpControle.Size = new System.Drawing.Size(244, 77);
this.grpControle.TabIndex = 8;
this.grpControle.TabStop = false;
this.grpControle.Text = "Controle";
//
// rdManual
//

```

```

this.rdManual.AutoSize = true;
this.rdManual.Checked = true;
this.rdManual.Location = new System.Drawing.Point(15, 35);
this.rdManual.Name = "rdManual";
this.rdManual.Size = new System.Drawing.Size(60, 17);
this.rdManual.TabIndex = 1;
this.rdManual.TabStop = true;
this.rdManual.Text = "Manual";
this.rdManual.UseVisualStyleBackColor = true;
this.rdManual.Click += new
System.EventHandler(this.rdManual_Click);
//
// rdAutomatico
//
this.rdAutomatico.AutoSize = true;
this.rdAutomatico.Location = new System.Drawing.Point(133, 35);
this.rdAutomatico.Name = "rdAutomatico";
this.rdAutomatico.Size = new System.Drawing.Size(78, 17);
this.rdAutomatico.TabIndex = 0;
this.rdAutomatico.Text = "Automático";
this.rdAutomatico.UseVisualStyleBackColor = true;
this.rdAutomatico.Click += new
System.EventHandler(this.rdAutomatico_Click);
//
// lblUmidade
//
this.lblUmidade.AutoSize = true;
this.lblUmidade.Location = new System.Drawing.Point(3, 16);
this.lblUmidade.Name = "lblUmidade";
this.lblUmidade.Size = new System.Drawing.Size(163, 13);
this.lblUmidade.TabIndex = 9;
this.lblUmidade.Text = "Defina a faixa de umidade (0 a 7)";
//
// MinUmidade
//
this.MinUmidade.Location = new System.Drawing.Point(6, 46);
this.MinUmidade.Name = "MinUmidade";
this.MinUmidade.Size = new System.Drawing.Size(45, 20);
this.MinUmidade.TabIndex = 10;
//
// label2

```

```

//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(57, 53);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(13, 13);
this.label2.TabIndex = 11;
this.label2.Text = "a";
//
// MaxUmidade
//
this.MaxUmidade.Location = new System.Drawing.Point(76, 46);
this.MaxUmidade.Name = "MaxUmidade";
this.MaxUmidade.Size = new System.Drawing.Size(45, 20);
this.MaxUmidade.TabIndex = 12;
//
// pnUmidade
//
this.pnUmidade.Controls.Add(this.lblUmidade);
this.pnUmidade.Controls.Add(this.MaxUmidade);
this.pnUmidade.Controls.Add(this.MinUmidade);
this.pnUmidade.Controls.Add(this.label2);
this.pnUmidade.Location = new System.Drawing.Point(12, 136);
this.pnUmidade.Name = "pnUmidade";
this.pnUmidade.Size = new System.Drawing.Size(200, 100);
this.pnUmidade.TabIndex = 13;
this.pnUmidade.Visible = false;
//
// textTeste
//
this.textTeste.Location = new System.Drawing.Point(409, 201);
this.textTeste.Name = "textTeste";
this.textTeste.Size = new System.Drawing.Size(117, 20);
this.textTeste.TabIndex = 14;
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(337, 248);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(157, 13);
this.label3.TabIndex = 15;

```

```

this.label3.Text = "Daniel dos Santos Ferreira Neto";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(337, 272);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(129, 13);
this.label4.TabIndex = 16;
this.label4.Text = "Prof. M.C Claudio Penedo";
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(538, 351);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.textTeste);
this.Controls.Add(this.pnUmidade);
this.Controls.Add(this.grpControle);
this.Controls.Add(this.lblStatus);
this.Controls.Add(this.btndesligar);
this.Controls.Add(this.txtLeitura);
this.Controls.Add(this.label1);
this.Controls.Add(this.btnligar);
this.Name = "Form1";
this.Text = "Form1";
this.grpControle.ResumeLayout(false);
this.grpControle.PerformLayout();
this.pnUmidade.ResumeLayout(false);
this.pnUmidade.PerformLayout();
this.ResumeLayout(false);
this.PerformLayout();
}
#endregion
private System.Windows.Forms.Button btnligar;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.TextBox txtLeitura;
private System.Windows.Forms.Button btndesligar;
private System.Windows.Forms.Timer timer1;

```

```

private System.Windows.Forms.Label lblStatus;
private System.Windows.Forms.GroupBox grpControle;
private System.Windows.Forms.RadioButton rdManual;
private System.Windows.Forms.RadioButton rdAutomatico;
private System.Windows.Forms.Label lblUmidade;
private System.Windows.Forms.TextBox MinUmidade;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox MaxUmidade;
private System.Windows.Forms.Panel pnUmidade;
private System.Windows.Forms.TextBox textTeste;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
    }
}

```

MODO MANUAL E AUTOMÁTICO

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace SisIrigacao
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        string Status;
        string Umidade;

        private void button1_Click(object sender, EventArgs e)
        {
            //LigaIrigacao;
            Procedimento.LigaIrigacao();
        }
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    //Desliga Irrigação
    Procedimento.DeligaIrigacao();
}
private void timer1_Tick(object sender, EventArgs e)
{
    string byteLeitura = string.Empty;
    byte A = Paralela.Ler(Convert.ToInt32("379", 16));

    byteLeitura = Convert.ToString(A, 2);

    while (byteLeitura.Length < 8)
        byteLeitura = "0" + byteLeitura;

    // Verifica o Status do Sistema
    if (byteLeitura.Substring(4,1).ToString().Trim() == "1")
        Status = "Sistema Ligado";
    else
        Status = "Sistema Desligado";

    if ((byteLeitura.Substring(1, 3).ToString().Trim()) == "000")
        Umidade = "0";
    else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"001")
        Umidade = "1";
    else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"010")
        Umidade = "2";
    else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"011")
        Umidade = "3";
    else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"100")
        Umidade = "4";
    else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"101")
        Umidade = "5";
    else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"110")
        Umidade = "6";

```

```

        else if ((byteLeitura.Substring(1, 3).ToString().Trim()) ==
"111")

            Umidade = "7";

        // verifica se o sistema está no modo automático
        if (rdAutomatico.Checked == true)
        {
            if (((String.IsNullOrEmpty(MinUmidade.Text)) &&
(!String.IsNullOrEmpty(MaxUmidade.Text)))
            {
                if (int.Parse(MinUmidade.Text.Trim()) <=
int.Parse(MaxUmidade.Text.Trim()))
                {
                    if (int.Parse(Umidade.Trim()) <
int.Parse(MinUmidade.Text.Trim()))
                    {
                        if (Status != "Sistema Ligado")
                            Procedimento.LigaIrrigacao();
                    }
                    else if (int.Parse(Umidade.Trim()) >
int.Parse(MaxUmidade.Text.Trim()))
                    {
                        if (Status != "Sistema Desligado")
                            Procedimento.DeligaIrrigacao();
                    }
                }
            }
            else
            {
                MinUmidade.Text = String.Empty;
                MaxUmidade.Text = String.Empty;
                MessageBox.Show("A umidade mínima não pode ser maior
que a máxima!!!");
            }
        }

        lblStatus.Text = Status;
        txtLeitura.Text = Umidade;
        textTeste.Text = byteLeitura.Substring(1, 3).ToString().Trim();
    }

    private void rdAutomatico_Click(object sender, EventArgs e)
    {

```



```

        pnUmidade.Visible = true;
        btndesligar.Visible = false;
        btnligar.Visible = false;
    }

    private void rdManual_Click(object sender, EventArgs e)
    {
        pnUmidade.Visible = false;
        btndesligar.Visible = true;
        btnligar.Visible = true;
    }
}

```

ROTINA DE LEITURA E ESCRITA NA PORTA PARALELA

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Runtime.InteropServices;

namespace SisIrigacao
{
    class Paralela
    {
        // Escreve um byte no endereço
        [DllImport("Inpout32.dll", EntryPoint = "Out32")]
        public static extern void Escrever(int endereco, byte valor);

        // Lê um byte do endereço
        [DllImport("Inpout32.dll", EntryPoint = "Inp32")]
        public static extern byte Ler(int endereco);
    }
}

```

ROTINA LIGAR E DESLIGAR SISTEMA DE IRRIGAÇÃO

```

using System;
using System.Collections.Generic;
using System.Text;

```

```

using System.Windows.Forms;

namespace SisIrigacao
{
    public class Procedimento
    {

        public static void LigaIrigacao()
        {
            try
            {
                int endereco = Convert.ToInt32("378", 16);
                byte dados = Convert.ToByte("00000001", 2);
                Paralela.Escrever(endereco, dados);

                System.Threading.Thread.Sleep(2000); // Permite o envio do
sinal por 2 segundos
                dados = Convert.ToByte("00000011", 2);
                Paralela.Escrever(endereco, dados);
            }
            catch (Exception ex)
            {
                throw new Exception("Erro ao ligar a irrigação", ex);
            }
        }

        public static void DeligaIrigacao()
        {
            try
            {
                int endereco = Convert.ToInt32("378", 16);
                byte dados = Convert.ToByte("00000000", 2);
                Paralela.Escrever(endereco, dados);

                System.Threading.Thread.Sleep(2000); // Permite o envio do
sinal por 2 segundos
                dados = Convert.ToByte("00000010", 2);
                Paralela.Escrever(endereco, dados);
            }
            catch (Exception ex)
            {
                throw new Exception("Erro ao desligar a irrigação", ex);
            }
        }
    }
}

```

```

    }
}
}

```

CODIGO DA FUNÇÃO MAIN

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace SisIrigacao
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```