ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are integral to critical infrastructure such as energy, water, and transportation networks. However, their increasing connectivity to public and private networks has made them vulnerable to Distributed Denial of Service (DDoS) attacks. This paper proposes a machine learning-based security mechanism using a decision tree algorithm to detect and mitigate DDoS threats in SCADA systems. The proposed approach analyzes network traffic patterns and classifies anomalies in real-time, enhancing system resilience without compromising performance. Simulation results demonstrate the effectiveness of the decision tree in accurately identifying malicious traffic with high detection rates and minimal false positives.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

SCADA systems are at the heart of industrial automation and critical infrastructure control, facilitating real-time monitoring and control of physical processes. With the adoption of IP-based communication protocols and increased internet connectivity, these systems have become more exposed to cyber threats, particularly Distributed Denial of Service (DDoS) attacks. DDoS attacks aim to overwhelm system resources, leading to significant operational disruption, safety risks, and economic loss.Traditional intrusion detection systems are often ineffective in dynamic SCADA environments due to their rigid rule-based structure. Therefore, there is a growing need for intelligent, adaptive methods to detect and respond to such attacks. Machine learning offers promising techniques for automated detection of abnormal traffic patterns. Among various algorithms, the decision tree classifier is notable for its simplicity, interpretability, and fast processing capabilities—making it well-suited for SCADA systems with real-time constraints.This study explores the application of a decision tree-based model to detect DDoS attacks targeting SCADA systems. It outlines the methodology for feature extraction, model training, and real-time deployment to classify legitimate and malicious traffic efficiently.

Supervisory Control and Data Acquisition (SCADA) systems play a critical role in monitoring and controlling industrial processes across various sectors such as energy, water treatment, manufacturing, and transportation. Due to their increasing connectivity with public and private networks, these systems have become vulnerable to a range of cyber threats, with Distributed Denial of Service (DDoS) attacks being among the most disruptive. A successful DDoS attack can cripple SCADA operations by overwhelming network resources, delaying crucial control commands, or completely shutting down communication channels.

Traditional cybersecurity solutions are often insufficient for SCADA systems due to their real-time requirements and limited computational resources compromising system performance.

1.1 LITERATURE SURVEY

1. NIST SP 800-82 – Guide to Industrial Control Systems (ICS) Security

This document provides guidance on how to secure Industrial Control Systems (ICS), including SCADA systems, against cybersecurity threats. It outlines common ICS architectures, identifies unique threats, and offers specific recommendations for securing these systems. The guide is designed to help organizations reduce risk by applying security controls tailored to ICS operations. ICS are critical to national security and economic stability, operating vital infrastructure such as power, water, and manufacturing systems. With increased interconnectivity, these systems face growing cyber threats. NIST SP 800-82 introduces the need for security in ICS environments and serves as a foundational reference for understanding threat vectors and risk mitigation strategies.

2. Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015) – Intrusion Detection and Big Heterogeneous Data

This survey presents an overview of intrusion detection techniques utilizing machine learning to process and analyze big heterogeneous datasets. It discusses different approaches, challenges in processing high-dimensional data, and evaluates the strengths and weaknesses of supervised and unsupervised models, including decision trees.As cyberattacks become increasingly sophisticated, traditional security methods fall short. The paper motivates the use of machine learning for intrusion detection, particularly in environments generating large and diverse data, like SCADA systems. It outlines the challenges in extracting patterns from such data and the role of classifiers, such as decision trees, in improving detection accuracy.

3. Milani Comparetti, P., et al. (2009) – A Host-based Intrusion Detection System with Decision Trees for SCADA Systems

This paper presents a host-based intrusion detection system (HIDS) tailored for SCADA systems. The system uses decision tree classifiers to identify abnormal behavior indicative of cyberattacks. Experimental results demonstrate the system's capability to detect various attack types while maintaining performance suitable for real-time operations. SCADA systems are increasingly targeted by cyber attackers due to their growing exposure and critical importance. Traditional IDS tools are often unsuitable for SCADA due to their lightweight design and real-time constraints.

4. Shao, H., & Zhao, J. (2016) – Anomaly Detection Model for SCADA System Based on Machine Learning

The authors propose an anomaly detection framework for SCADA systems based on machine learning algorithms. By analyzing communication traffic and system behavior, the model identifies deviations from normal operations. The paper compares multiple ML algorithms, highlighting the effectiveness of decision trees in detecting anomalies such as DDoS attacks. Modern SCADA systems are facing serious cyber threats, with DDoS attacks posing a major concern. This paper highlights the importance of timely and accurate detection of such anomalies. By employing ML techniques, the authors aim to automate the identification of threats and reduce false positives in real-time monitoring scenarios.

5. Ali, T., et al. (2018) – Detection of DDoS Attacks Using Machine Learning Techniques in SCADA Environment

This study investigates the application of machine learning algorithms to detect DDoS attacks in SCADA environments. Using a simulated testbed, data was collected and used to train several classifiers, including decision trees. Results indicate that decision trees provide high accuracy and low false alarm rates, making them suitable for real-time defense mechanisms. SCADA systems are increasingly vulnerable to DDoS attacks due to their open communication protocols and lack of built-in security. The paper introduces the concept of using machine learning to enhance intrusion detection capabilities in these environments. It further justifies the use of decision trees due to their interpretability and speed, especially in resource-constrained SCADA systems.

1.2 EXISTING SYSTEM

Existing methods for securing SCADA (Supervisory Control and Data Acquisition) systems against Distributed Denial of Service (DDoS) attacks primarily rely on traditional network security mechanisms such as firewalls, intrusion detection systems (IDS), and rule-based filtering. While these approaches offer some level of protection, they are often not sufficient to handle the unique challenges posed by SCADA environments. Many SCADA systems were originally designed as isolated networks with minimal consideration for cybersecurity. As these systems become increasingly integrated with IT infrastructures and the Internet, they inherit vulnerabilities that make them susceptible to cyberattacks. Traditional IDS solutions, while effective in enterprise networks, often struggle to cope with the real-time and deterministic nature of SCADA protocols. They typically generate a high number of false positives or fail to detect sophisticated or low-and-slow DDoS attacks.

1.3 PROPOSED SYSTEM

To address the limitations of existing SCADA security mechanisms, this study proposes an intelligent and efficient approach to detect Distributed Denial of Service (DDoS) attacks using a **Decision Tree-based classification model**. The proposed system is designed to enhance threat detection accuracy while maintaining low computational overhead, making it suitable for deployment in real-time SCADA environments.

The core idea behind the proposed system is to leverage machine learning—specifically decision trees—to learn from historical network traffic data and distinguish between normal and malicious traffic patterns. Decision trees are selected due to their simplicity, fast processing speed, and interpretability, which are essential features for integration into SCADA systems with limited processing capabilities.

1.4 BLOCK DIAGRAM

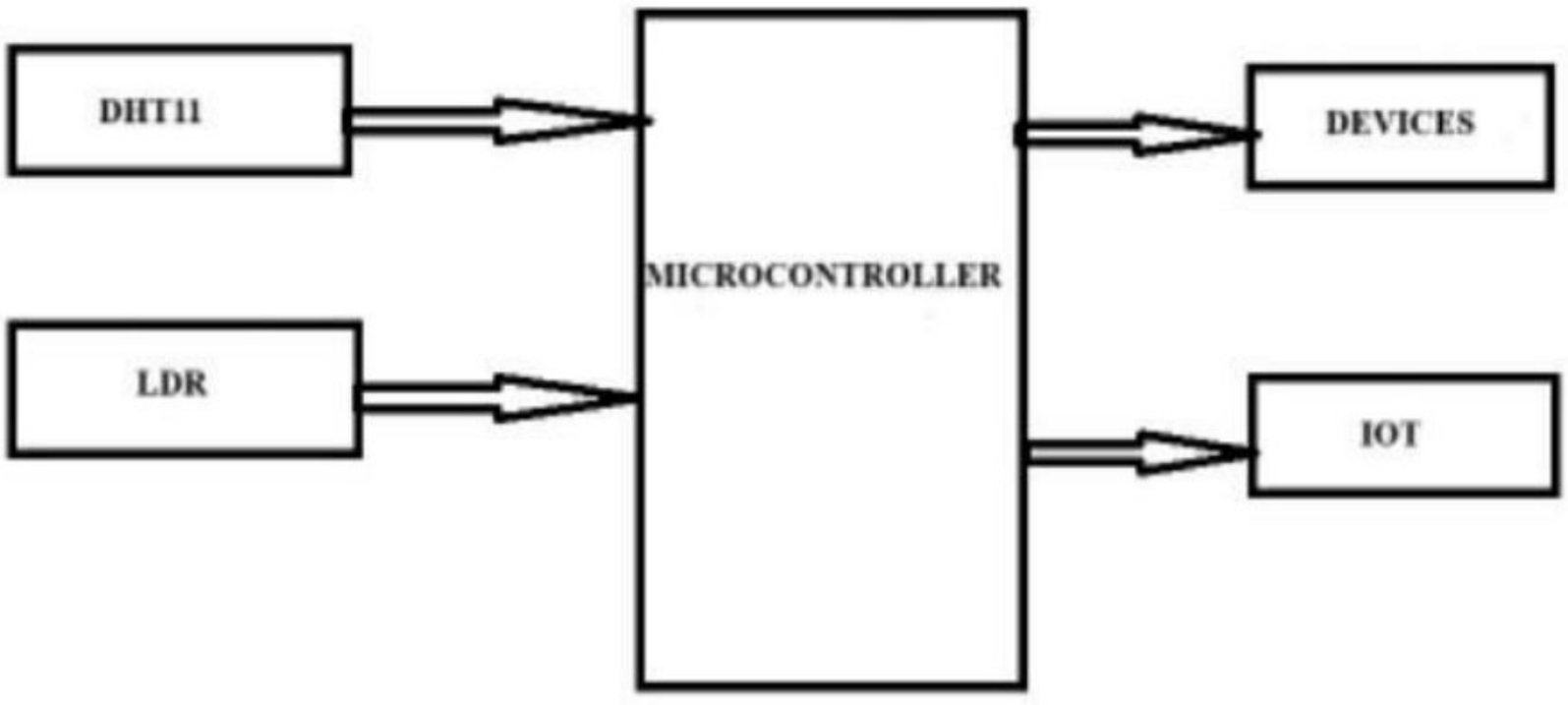


Figure 1.1 Securing the SCADA system from DDOS attacks identifying using a decision tree

- **LDR(Light Dependent Resistor):**

A **Light Dependent Resistor (LDR)**, also known as a **photoresistor**, is a type of resistor whose resistance varies significantly with the intensity of light falling on it.

- **DHT11:**

The **DHT11** is a **basic, low-cost digital sensor** used to measure **temperature and humidity**. It's widely used in DIY electronics, IoT projects, and embedded systems for environmental monitoring.

- **IOT:**

The **Internet of Things (IoT)** refers to a network of **physical devices**—such as sensors, actuators, appliances, vehicles, and machines—that are connected to the **internet** and capable of **collecting, sharing, and acting on data**.

CHAPTER 2

INTRODUCTION TO EMBEDDED SYSTEMS

Many embedded systems have substantially different design constraints than desktop computing applications. No single characterization applies to the diverse spectrum of embedded systems. However, some combination of cost pressure, long life-cycle, real-time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer design methodologies and tools to embedded applications. Embedded systems in many cases must be optimized for life-cycle and business-driven factors rather than for maximum computing throughput. There is currently little *tool* support for expanding embedded computer design to the scope of holistic embedded system design. However, knowing the strengths and weaknesses of current approaches can set expectations appropriately, identify risk areas to tool adopters, and suggest ways in which tool builders can meet industrial needs.

If we look around us, today we see numerous appliances which we use daily, be it our refrigerator, the microwave oven, cars, PDAs etc. Most appliances today are powered by something beneath the sheath that makes them do what they do. These are tiny microprocessors, which respond to various keystrokes or inputs. These tiny microprocessors, working on basic assembly languages, are the heart of the appliances. We call them embedded systems. Of all the semiconductor industries, the embedded systems market place is the most conservative, and engineering decisions here usually lean towards established, low risk solutions. Welcome to the world of embedded systems, of computers that will not look like computers and won't function like anything we are familiar with.

2.1 CLASSIFICATION

Embedded systems are divided into autonomous, real-time, networked & mobile categories.

- **Autonomous systems**

  They function in standalone mode. Many embedded systems used for process control in manufacturing units& automobiles fall under this category.

- **Real-time embedded systems**

  These are required to carry out specific tasks in a specified amount of time. These systems are extensively used to carry out time critical tasks in process control.

- **Networked embedded systems**

  They monitor plant parameters such as temperature, pressure and humidity and send the data over the network to a centralized system for on line monitoring.

- **Mobile gadgets**

  Mobile gadgets need to store databases locally in their memory. These gadgets imbibe powerful computing & communication capabilities to perform real time as well as nonrealtime tasks and handle multimedia applications. The embedded system is a combination of computer hardware, software, firmware and perhaps additional mechanical parts, designed to perform a specific function. A good example is an automatic washing machine or a microwave oven. Such a system is in direct contrast to a personal computer, which is not designed to do only a specific task. But an embedded system is designed to do a specific task with in a given timeframe, repeatedly, endlessly, with or without human interaction.

- **Hardware**

  Good software design in embedded systems stems from a good understanding of the hardware behind it. All embedded systems need a microprocessor, and the kinds of microprocessors used in them are quite varied. A list of some of the common microprocessors families are: ARM family, The Zilog Z8 family, Intel 8051/X86 family, Motorola 68K family and the power PC family. For processing of information and execution of programs, embedded system incorporates microprocessor or micro-controller. In an embedded system the microprocessor is a part of final product and is not available for reprogramming to the end user. An embedded system also needs memory for two purposes, to store its program and to store its data. Unlike normal desktops in which data and programs are stored at the same place, embedded systems store data and programs in different memories. This is simply because the embedded system does not have a hard drive and the program must be stored in memory even when the power is turned

CHAPTER 3

N0DE-MCU

3.1 ARDUINO

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an ATmega microcontroller essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts. The Arduino connects to your computer via USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the Arduino can run with the USB link back to your computer, or stand-alone without it no keyboard or screen needed, just power.

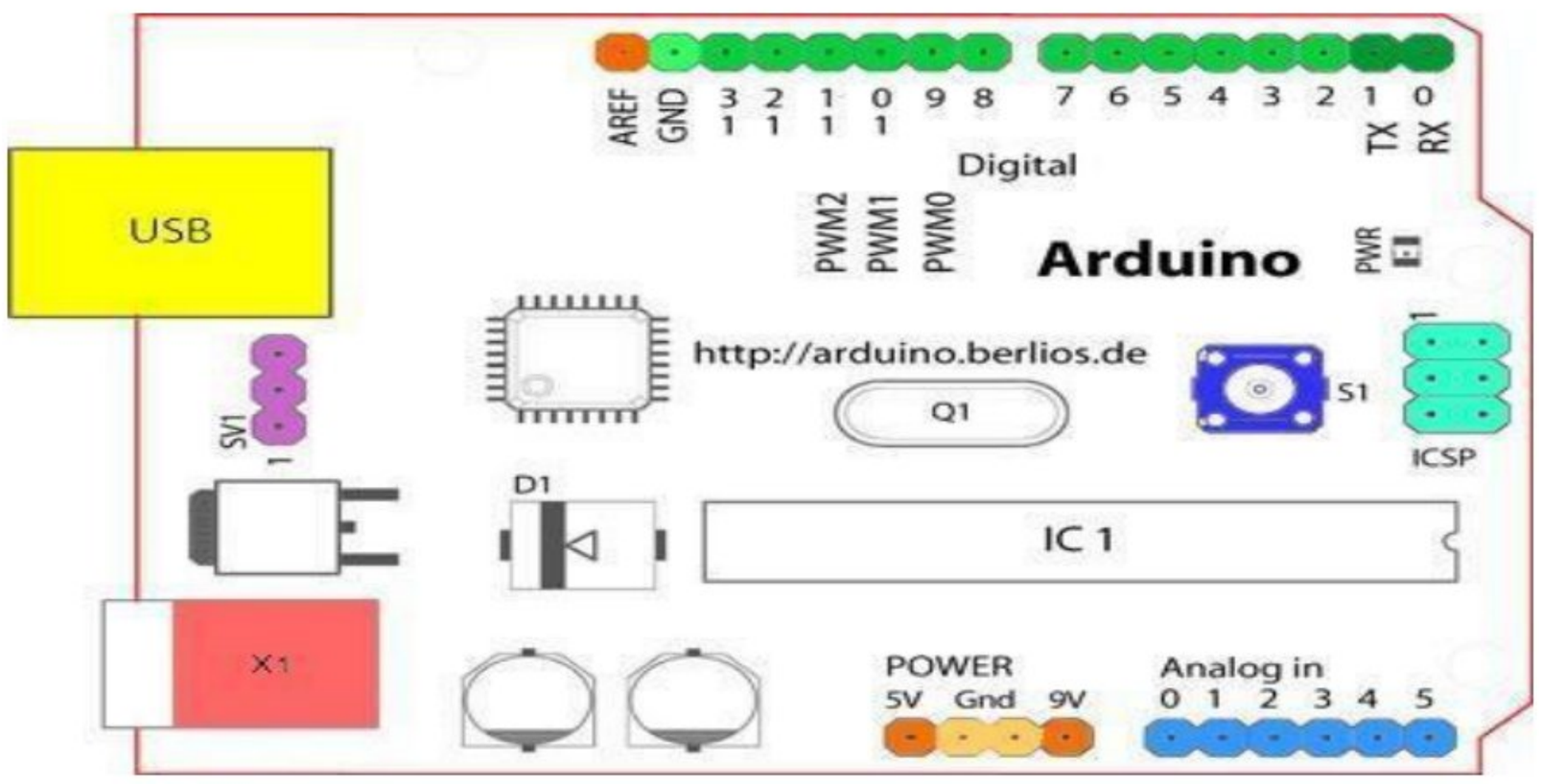


Figure 3.1 Smart Home Warning system

Looking at the board from the top down, this is an outline of what you will see (parts of the board you might interact with in the course of normal use are highlighted).

Figure 3.2Arduino Board

Starting clockwise from the top center

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (Digital Read and Digital Write) if you are also using serial communication (e.g. Serial.begin).
- Reset Button - S1 (dark blue)

- In-circuit Serial Programmer (blue-green)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

3.2 DIGITAL PINS

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pin Mode(), Digital Read(), and Digital Write() commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40mA.

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11** Provide 8-bit PWM output with the analog Write() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- **BT Reset: 7.** (Arduino BT-only) Connected to the reset line of the bluetooth module.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

3.3 ANALOG PINS

In addition to the specific functions listed below, the analog input pins support 10-bitanalog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

- **I^2C: 4 (SDA) and 5 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website).

3.4 POWER PINS

- **VIN** (sometimes labeled "9V"): The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Also note that the Lily Pad has no VIN pin and accepts only a regulated input.

- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- **3V3** (Diecimila-only) : A 3.3 volt supply generated by the on-board FTDI chip.

- **GND:** Ground pins.

3.5 OTHER PINS

- **AREF:** Reference voltage for the analog inputs. Used with analog Reference().
- **Reset:** (Decimal-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

3.6 ATMEGA328

Pin diagram

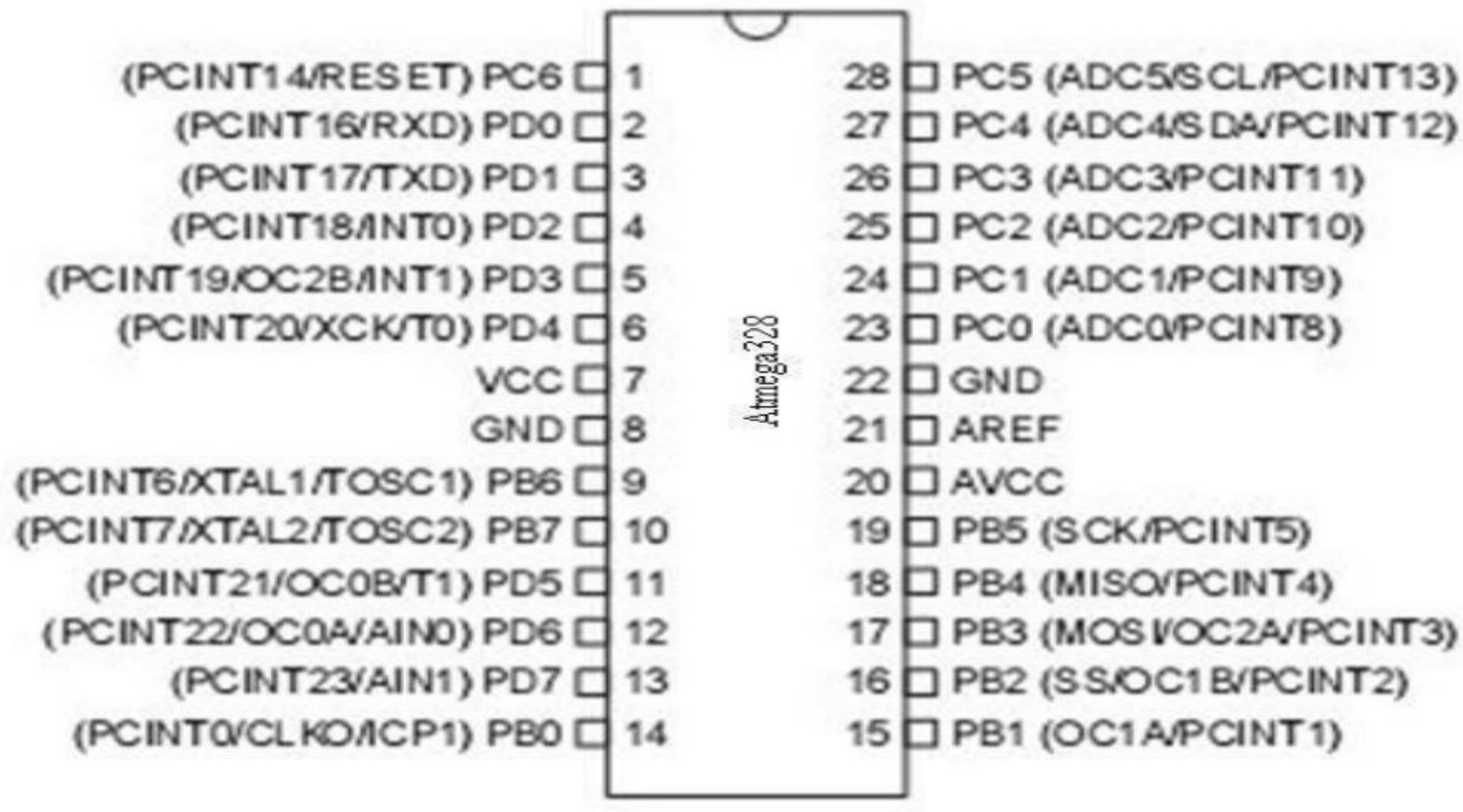


Figure 3.3 Pin Configuration of Atmega328

Pin Description

- **VCC:**

  Digital supply voltage.

- **GND:**

  Ground.

- **Port A (PA7-PA0):**

Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running

- **Port B (PB7-PB0):**

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega32.

- **Port C (PC7-PC0):**

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TD0 pin is tri-stated unless TAP states that shift out data are entered. Port C also serves the functions of the JTAG interface.

- **Port D (PD7-PD0):**

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega32.

- **Reset (Reset Input):**

A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

- **XTAL1:**

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

- **XTAL2:**

Output from the inverting Oscillator amplifier.

- **AVCC:**

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

- **AREF:**

AREF is the analog reference pin for the A/D Converter

3.7 FEATURES

- 1.8-5.5V operating range
- Up to 20MHz
- Part: ATMEGA328P-AU
- 2 8-bit Timer/Counter
- 16-bit Timer/Counter
- RTC with separate oscillator
- 6 PWM Channels
- 8 Channel 10-bit ADC
- Serial USART
- 2-wire (I2C) interface
- Watchdog timer
- Analog comparator
- 23 IO lines
- Data retention: 20 years at 85C/ 100 years at 25C
- Digital I/O Pins are 14 (out of which 6 provide PWM output)
- Analog Input Pins are 6.
- DC Current per I/O is 40 mA
- DC Current for 3.3V Pin is 50mA
- 32kB Flash program

3.8 INTERRUPTS

This section describes the specifics of the interrupt handling as performed in the Atmega328. In Atmega328Each Interrupt Vector occupies two instruction words and the Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR.

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 0
4	0x0006	PCINTO	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter 2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter 2 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x001A	TIMER 1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A

16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIME0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART RX Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, TX Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

Table 3.1 Reset and Interrupt Vectors in ATMEGA 328 and ATMEGA 328P

When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.Table below shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

3.9 ARDUINO CHARACTERISTICS

3.9.1 Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- **IOREF:** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

3.9.2 Memory:

The ATmega328 has 32 KB (with 0.5 KB used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

3.9.3 Serial Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

CHAPTER 4

HARDWARE COMPONENTS

4.1 LCD (Liquid Crystal Display)

4.1.1 Introduction

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontrollers

4.1.2 Features:

- Interface with either 4-bit or 8-bit microprocessor.
- Display data RAM
- 80x8 bits (80 characters).
- Character generator ROM
- 160 different 5x7 dot-matrix character patterns.
- Character generator RAM
- 8 different user programmed 5x7 dot-matrix patterns.
- Display data RAM and character generator RAM may be Accessed by the microprocessor.
- Numerous instructions
- Clear Display, Cursor Home, Display ON/OFF, Cursor ON/OFF, Blink Character, Cursor Shift, Display Shift.
- Built-in reset circuit is triggered at power ON.
- Built-in oscillator.

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table 4.1 Address locations for a 1x16 line LCD

Figure 4.1 SHAPES AND SIZES

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

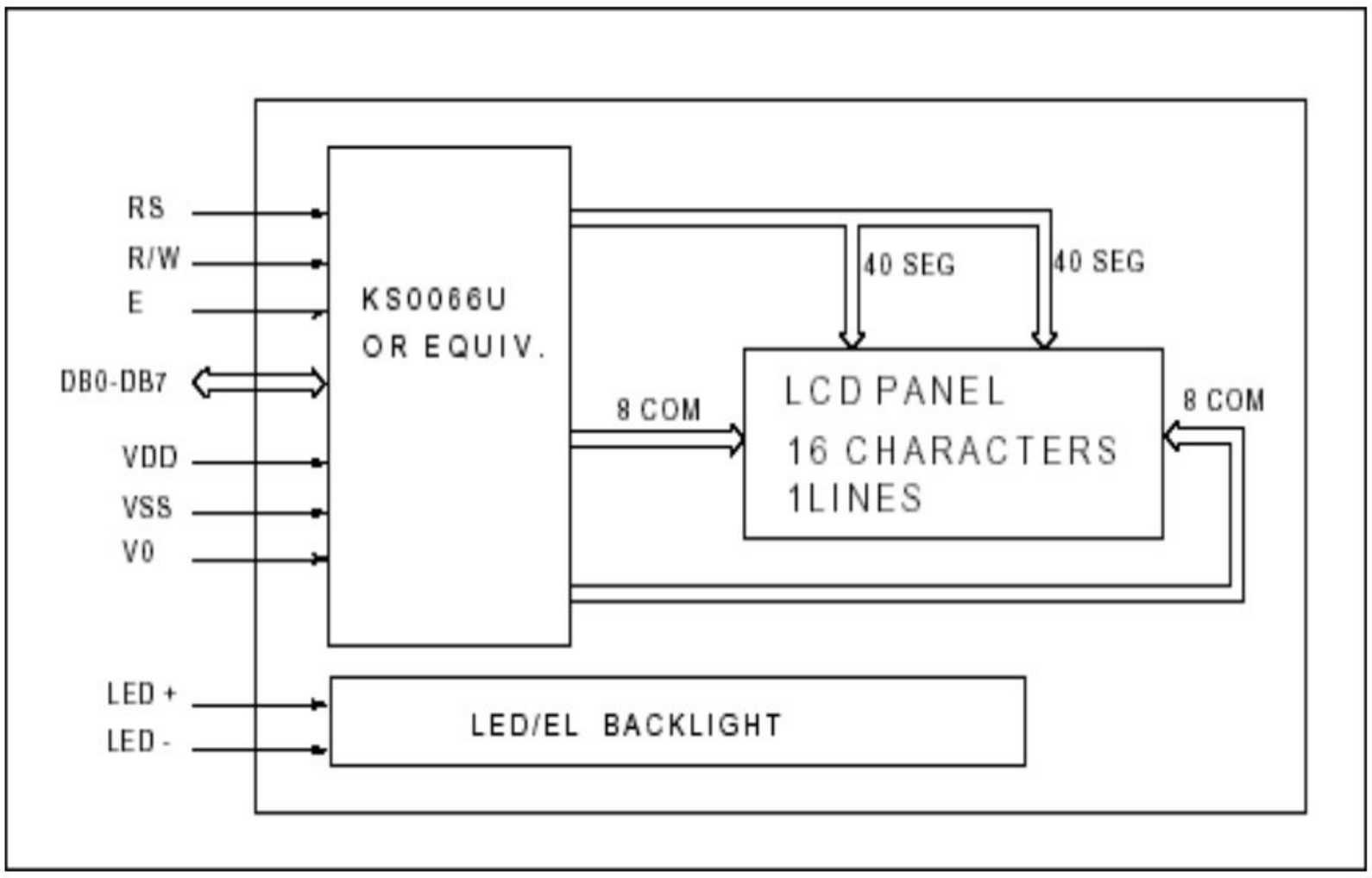


Figure 4.2 Electrical block diagram

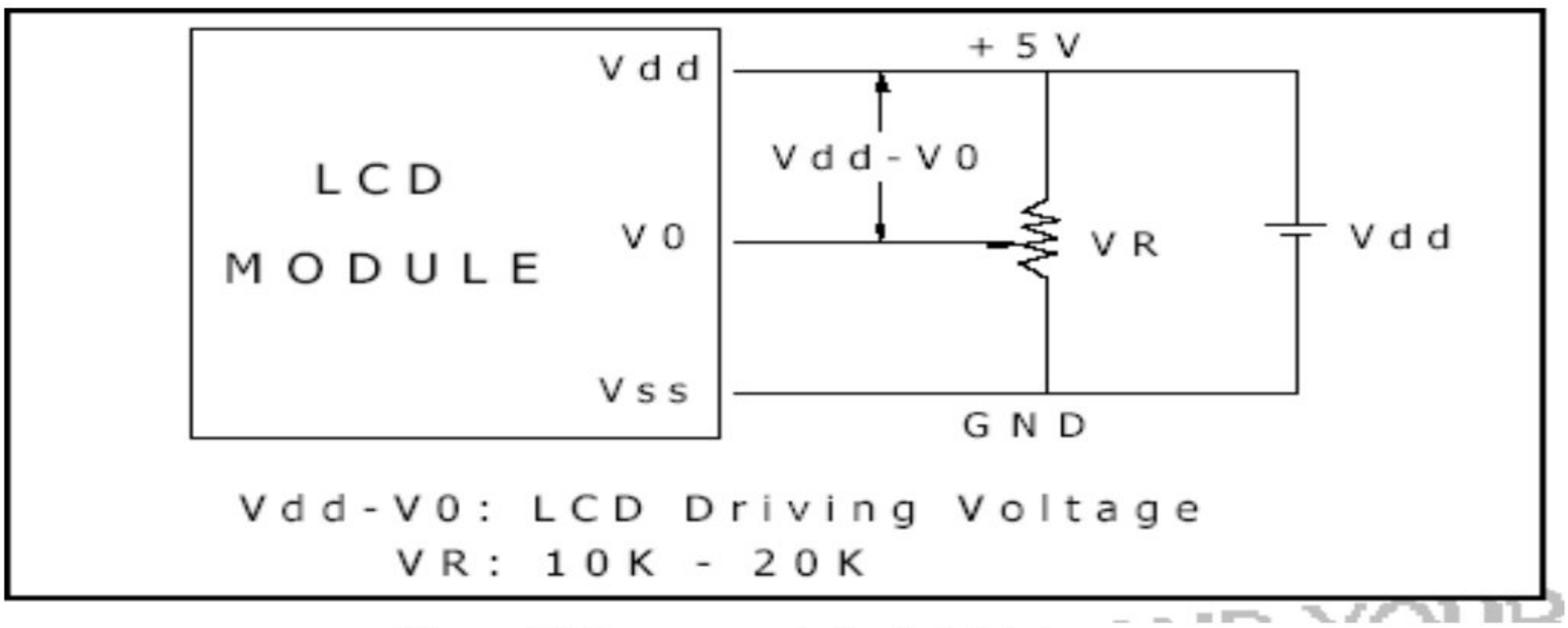


Figure 4.3 Power supply for lcd driving

4.1.3 PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

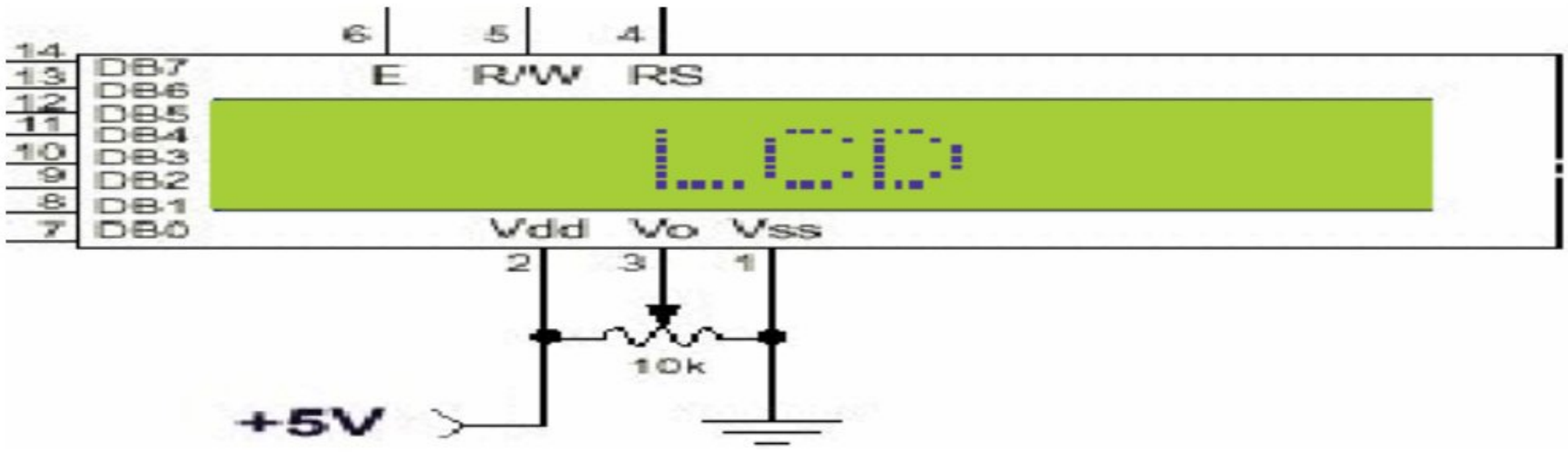


Figure 4.4 pin diagram of 1x16 lines lcd

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

4.1.4 CONTROL LINES:

- **EN:**

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required.

- **RS:**

Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which sould be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

- **RW:**

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

4.1.5 Logic status on control lines:

- E - 0 Access to LCD disabled, 1 Access to LCD enabled
- R/W - 0 Writing data to LCD, 1 Reading data from LCD
- RS - 0 Instructions
- Character

4.1.6 Writing data to the LCD:

- Set R/W bit to low
- Set RS bit to logic 0 or 1 (instruction or character)
- Set data to data lines (if it is writing)
- Set E line to high
- Set E line to low

4.1.7 Read data from data lines (if it is reading) on LCD:

- Set R/W bit to high
- Set RS bit to logic 0 or 1 (instruction or character)
- Set data to data lines (if it is writing)
- Set E line to high

4.2 REGULATED POWER SUPPLY:

4.2.1 Introduction

Power supply is a supply of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

A power supply may include a power distribution system as well as primary or secondary sources of energy such as

- Conversion of one form of electrical power to another desired form and voltage, typically involving converting AC line voltage to a well-regulated lower-voltage DC for electronic devices. Low voltage, low power DC power supply units are commonly integrated with the devices they supply, such as computers and household electronics.
- Batteries.
- Chemical fuel cells and other forms of energy storage systems.
- Solar power.
- Generators or alternators.

4.2.2 Block Diagram

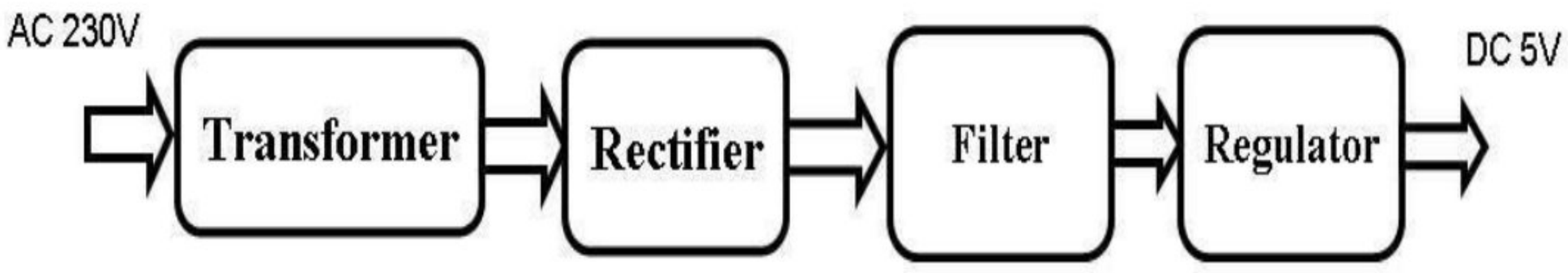


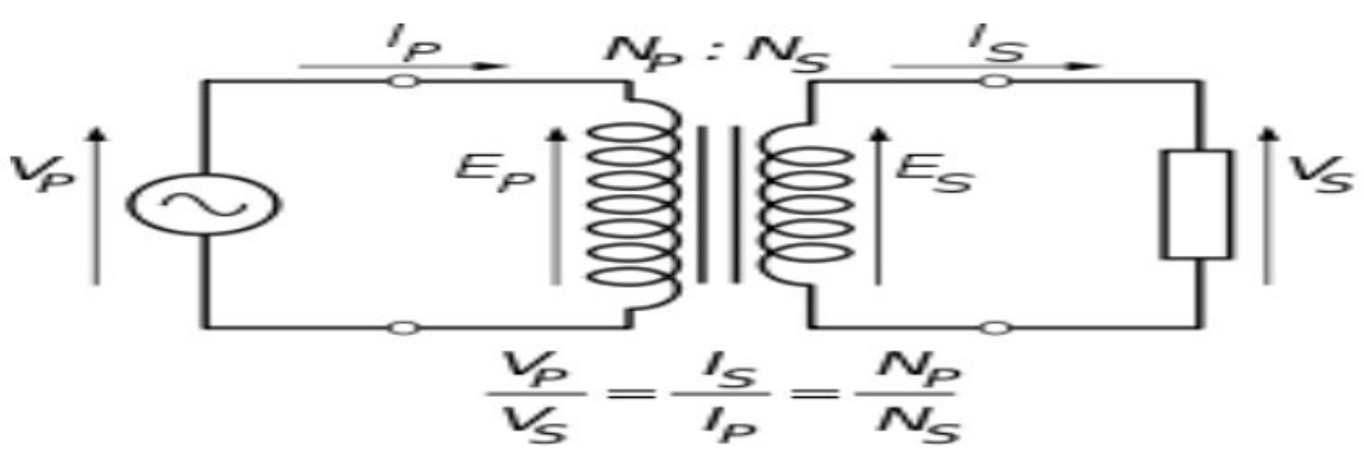
Figure 4.5 Regulated Power Supply

Step 1: Transformation:

The process of transforming energy from one device to another is called transformation. For transforming energy we use transformers.

- **Transformers:**

  A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequen.A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.



The voltage induced in the secondary is determined by the TURNS RATIO.

$$\frac{\text{primary voltage}}{\text{secondary voltage}} = \frac{\text{number of primary turns}}{\text{number of secondary turns}}$$

- **Step Up transformer:**

  In case of step up transformer, primary windings are every less compared to secondary winding Because of having more turns secondary winding accepts more energy, and it releases more voltage at the output side.

- **Step down transformer:**

  Incase of step down transformer, Primary winding induces more flux than the secondary winding, and secondary winding is having less number of turns because of that it accepts less number of flux, and releases less amount of voltage.

Step 2: Rectification

The process of converting an alternating current to a pulsating direct current is called as rectification. For rectification purpose we use rectifiers.

- **Rectifiers:**

A rectifier is an electrical device that converts alternating current (AC) to direct current (DC), a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components.

- **Bridge full wave rectifier:**

The Bridge rectifier circuit is shown in figure, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge.

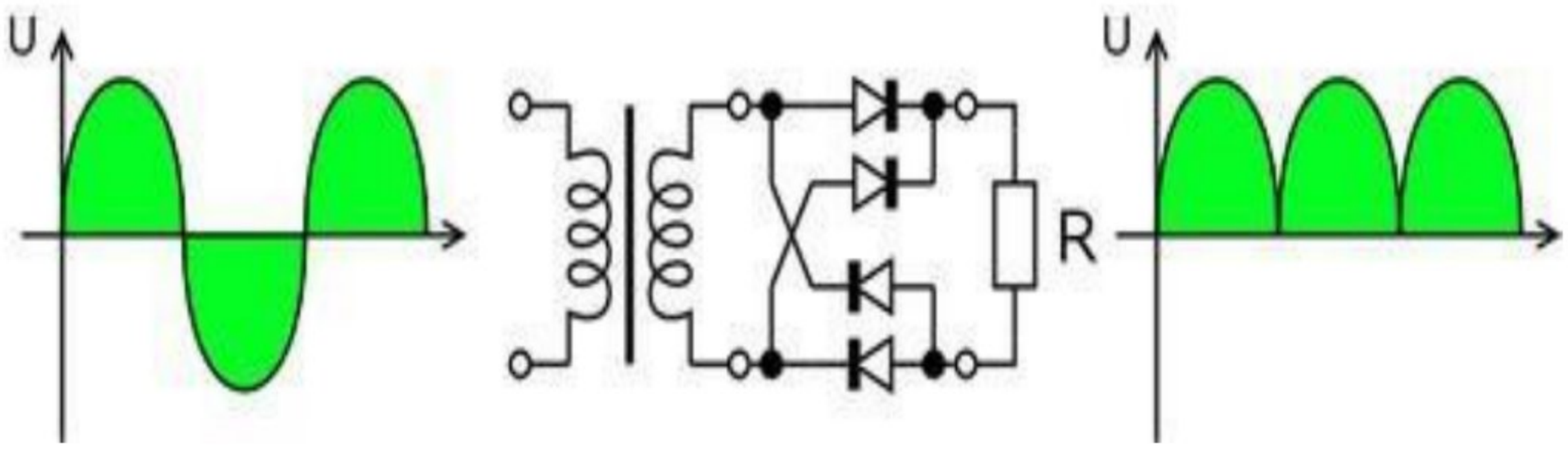


Figure 4.6 Bridge rectifier: a full-wave rectifier using 4 diodes

Step 3: Filtration

The process of converting a pulsating direct current to a pure direct current using filters is called as filtration.

- **Filters:**

Electronic filters are electronic circuits, which perform signal-processing functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones.

Step 4: Regulation

The process of converting a varying voltage to a constant regulated voltage is called as regulation. For the process of regulation we use voltage regulators.

- **Voltage Regulator:**

A voltage regulator (also called a 'regulator') with only three terminals appears to be a simple device, but it is in fact a very complex integrated circuit. It converts a varying input voltage into a constant 'regulated' output voltage. Voltage Regulators are available in a variety of outputs like 5V, 6V, 9V, 12V and 15V. The LM78XX series of voltage regulators are designed for positive input. For applications requiring negative input, the LM79XX series is used. Using a pair of 'voltage-divider' resistors can increase the output voltage of a regulator circuit.

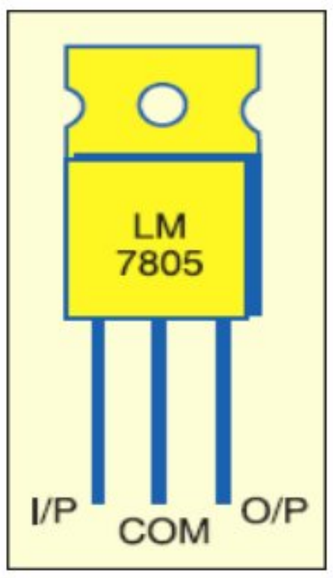


Figure 4.7 Voltage Regulator

4.3 LDR SENSOR

LDR (Light Dependent Resistor) as the name states is a special type of resistor that works on the photoconductivity principle means that resistance changes according to the intensity of light. Its resistance decreases with an increase in the intensity of light.

It is often used as a light sensor, light meter, Automatic street light, and in areas where we need to have light sensitivity. LDR is also known as a Light Sensor. LDR are usually available in 5mm, 8mm, 12mm, and 25mm dimensions.

4.3.1 Types of LDR or Photoresistors

1. Intrinsic Photoresistor

This type of photoresistor is made with pure semiconductors without any doping. This kind of photoresistor uses pure semiconductors like silicon and germanium. when the incident light with an adequate amount of energy falls on this, electrons gain that energy and get excited, and a few of them go to the conduction band.

2. Extrinsic Photoresistor

This type of photoresistor uses a doped semiconductor; this means some impurities are mixed with the semiconductor such as phosphorus to make this photoresistor.

Extrinsic light-dependent resistors are generally designed for longer wavelengths of light, with a tendency towards infrared (IR).

LDR Applications

- The photoresistor is generally used in detecting the presence and intensity of light
- Used in automatic lights that switch on and off according to light
- Clock with automatic light
- Optical circuit design
- Photo proximity switch
- Laser-based security systems

4.4 DHT TEMPERATURE & HUMIDITY SENSORS

These sensors are very basic and slow, but are great for hobbyists who want to do some basic data logging. The DHT sensors are made of two parts, a capacitive humidity sensor and a thermistor. There is also a very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity. The digital signal is fairly easy to read using any microcontroller.

Humidity is the measure of water vapour present in the air. The level of humidity in air affects various physical, chemical and biological processes. In industrial applications, humidity can affect the business cost of the products, health and safety of the employees.

So, in Semiconductor Industries and control system industries measurement of humidity is very important. Humidity measurement determines the amount of moisture present in the gas that can be a mixture of water vapour, nitrogen, argon or pure gas etc… Humidity sensors are of two types based on their measurement units. They are a relative humidity sensor and Absolute humidity sensor. DHT11 is a digital temperature and humidity sensor.

4.4.1DHT11 Sensor

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

4.4.2 DHT11 Applications

- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

4.5 SOFTWARE DESCRIPTION

4.5.1Arduino Software

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an ATmega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output

We need the following

4.5.1.1 A computer (Windows, Mac, or Linux)

4.5.1.2 An Arduino-compatible microcontroller (anything from this guide should work)

4.5.1.3 A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out this USB buying guide if you're not sure which cable to get).

Figure 4.8 Arduino Uno

Figure 4.9 USB Cable

4.5.1.4 An Arduino Uno

4.5.1.5 Windows 7, Vista, and XP

4.5.1.6 Installing the Drivers for the Arduino Uno (from Arduino.cc)

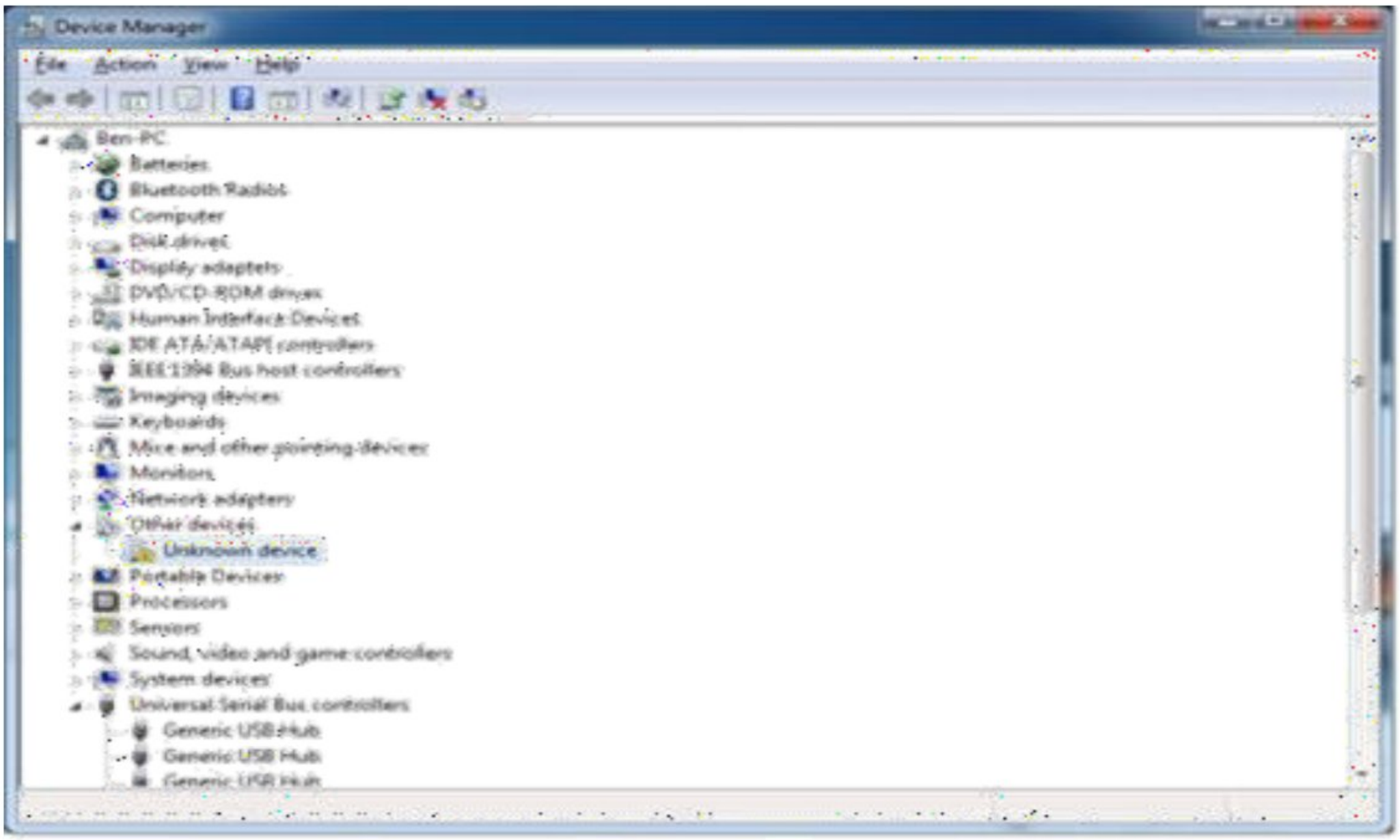
4.5.1.7 Plug in your board and wait for Windows to begin it's driver installation process After a

few moments, the process will fail, despite its best efforts

4.5.1.8 Click on the Start Menu, and open up the Control Panel

4.5.1.9 While in the Control Panel, navigate to System and Security. Next, click on System Once the System window is up, open the Device Manager

4.5.1.10 Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)".



4.5.1.11 If there is no COM & LPT section, look under 'Other Devices' for 'Unknown Device'

4.5.1.12 Right click on the "Arduino UNO (COMxx)" or "Unknown Device" port and choose the "Update Driver Software" opti Next, choose the "Browse my computer for Driver software" option.

4.5.1.13 Finally, navigate to and select the Uno's driver file, named "ArduinoUNO.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub- directory). If you cannot see the .inf file, it is probably just hidden. You can select the 'drivers' folder with the 'search sub-folders' option selected instead. Windows will finish up the driver installation

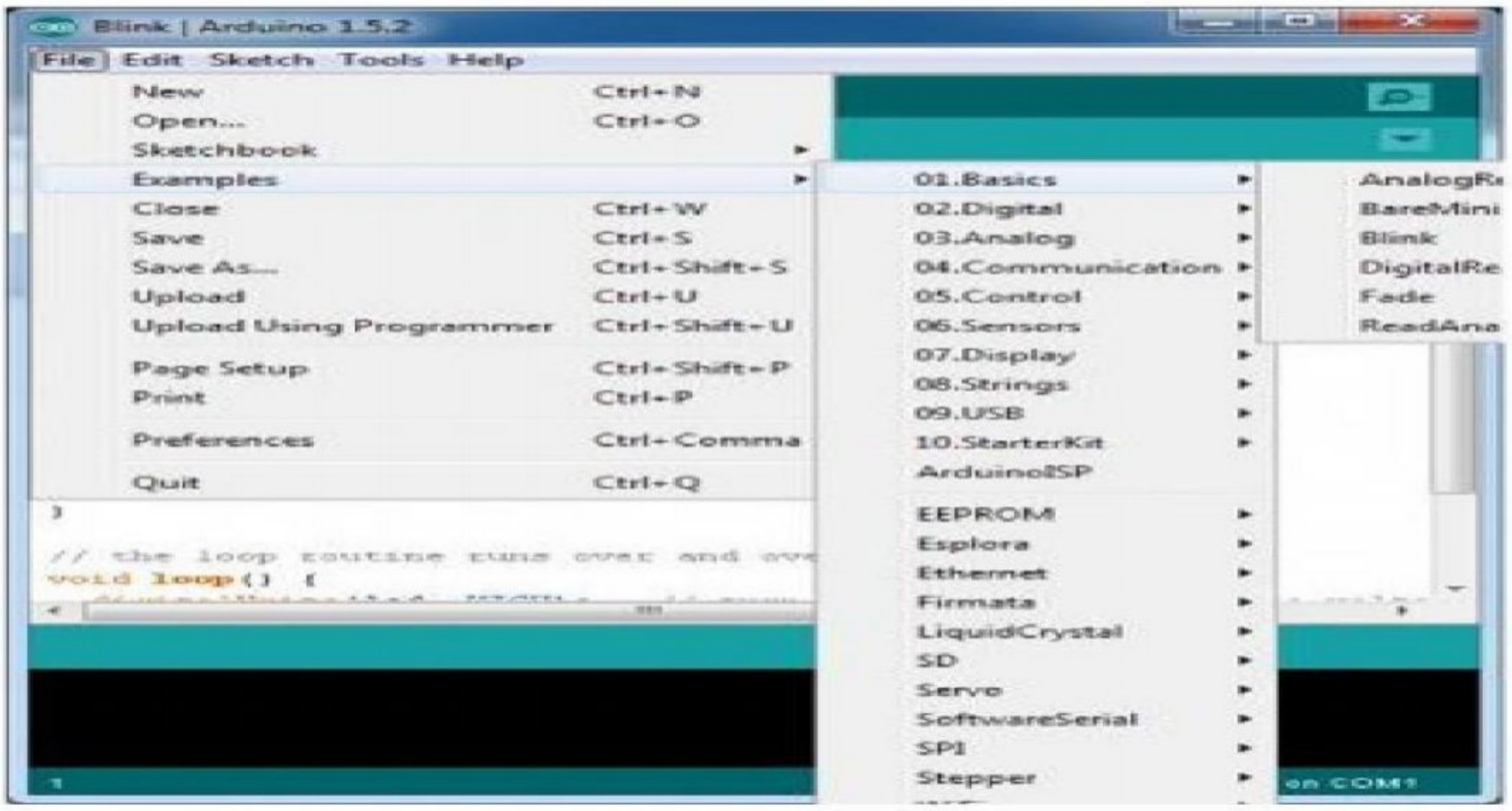
4.5.2 LAUNCH AND BLINK!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

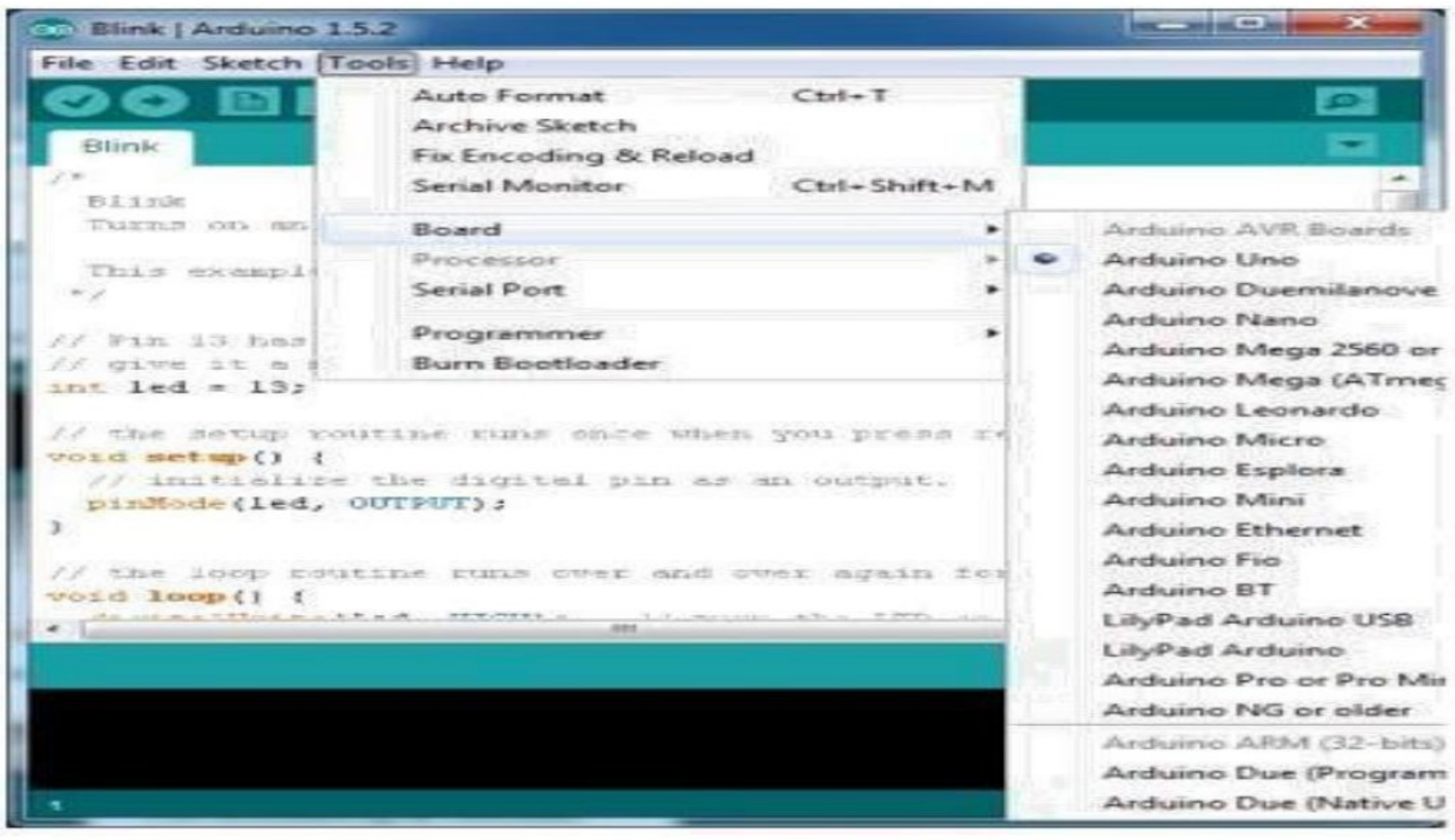
4.5.2.1 Launch the Arduino application

4.5.2.2 If you disconnected your board, plug it back in

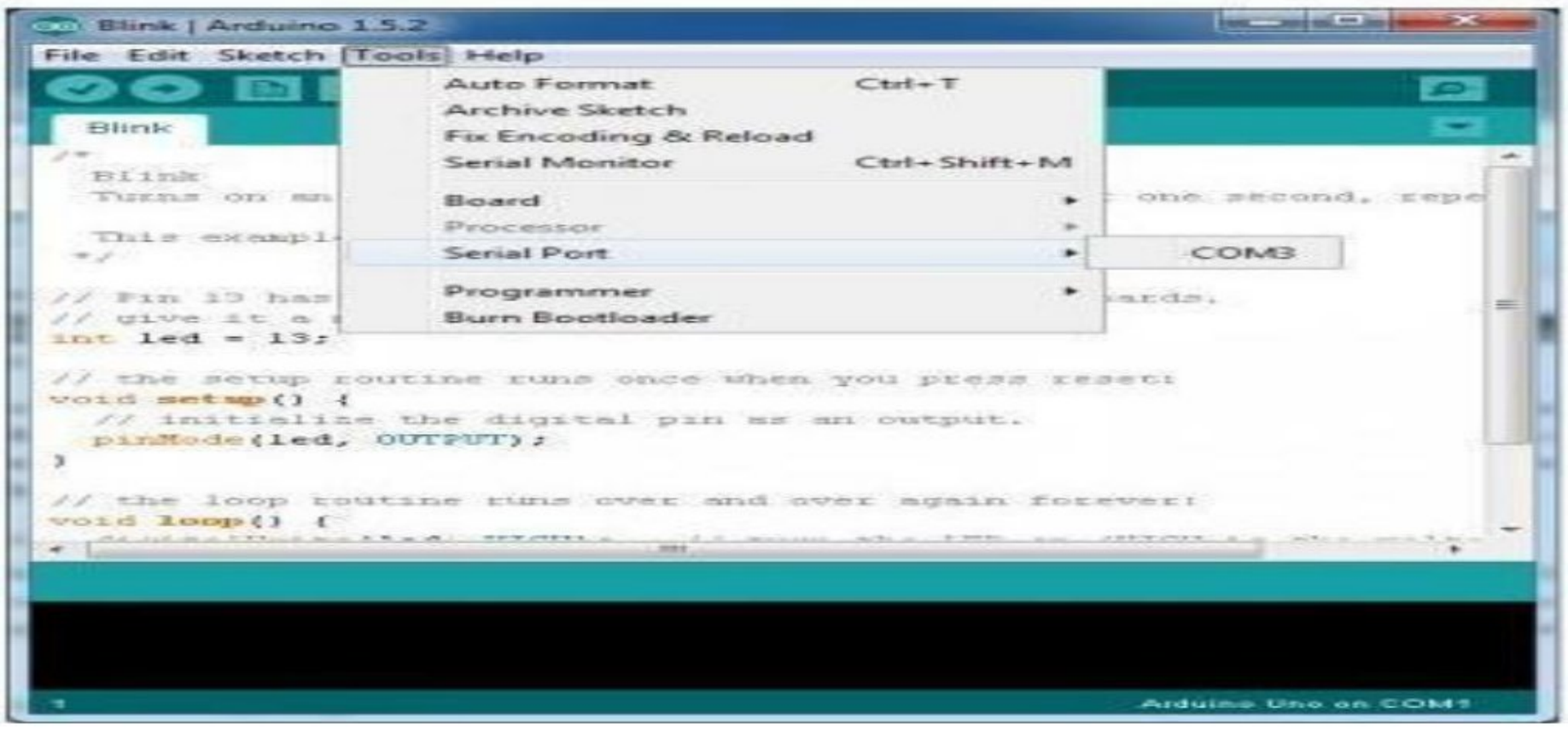
4.5.2.3 Open the Blink example sketch by going to: File > Examples > 1.Basics > Blink

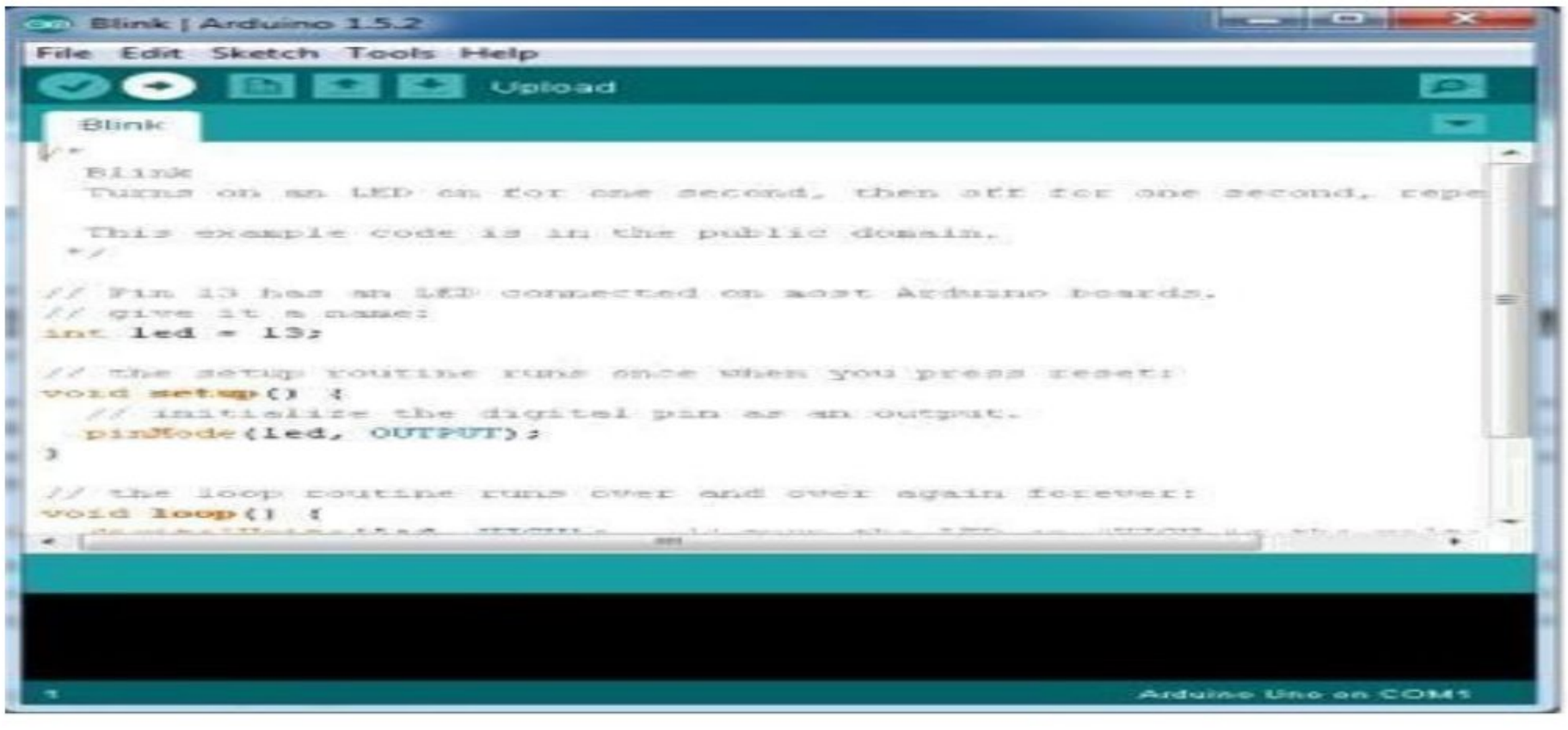


4.5.2.4 Select the type of Arduino board you're using: Tools > Board > your board type



4.5.2.5 Select the serial/COM port that your Arduino is attached to: Tools > Port > COMxx





If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino. With your Arduino board connected, and the Blink sketch open, press the 'Upload' button After a second, you should see some LEDs flashing on your Arduino, followed by the message 'Done Uploading' in the status bar of the Blink sketch. If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

CHAPTER 5

RESULT

Figure 4.10 Protype of securing SCADA system from DDOS attacks identifying using a decision tree

Inference

Securing a SCADA system from DDoS attacks can be achieved by employing a decision tree-based approach for intrusion detection. This approach can effectively identify and classify malicious traffic, allowing for proactive measures to mitigate the impact of DDoS attacks. Decision tree models, when used in conjunction with other methods like hybrid models or federated learning, can achieve high accuracy in detecting and classifying DDoS attacks in SCADA systems, especially in scenarios involving SDN-based SCADA systems.

CONCLUSION

In conclusion, Securing SCADA systems against DDoS attacks is crucial for maintaining the reliability and safety of critical infrastructure. This research presents a decision tree-based detection mechanism as an effective solution to identify and mitigate DDoS threats in real time. The experimental analysis confirms that decision trees provide a high level of accuracy and fast response times, making them suitable for deployment in resource-constrained SCADA environments. Future work may explore integrating ensemble models or deep learning techniques for enhanced detection capabilities while maintaining low computational overhead.

CHAPTER 6

REFERENCE

[1] Wang, P., Zhu, M., & Wang, J. (2017). "Detection of DDoS attack in SCADA system using machine learning techniques." *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, 1–5.

[2] Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). "Intrusion detection and Big Heterogeneous Data: a Survey." *Journal of Big Data*, 2(1), 3.

[3] Alam, M. M., & Vuong, S. T. (2016). "Scalable and distributed DDoS attack detection for cloud environments using convolutional neural networks." *Journal of Network and Computer Applications*, 97, 65–76.

[4] Rana, O., & Abdelrahman, O. H. (2020). "Machine learning approach to detect and prevent DDoS attacks in SCADA networks." *Procedia Computer Science*, 170, 1013–1020.

[5] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). "A survey of network anomaly detection techniques." *Journal of Network and Computer Applications*, 60, 19–31.

[6] Kandekar, M., & Arora, A. (2019). "Securing SCADA systems from DDoS attacks using supervised machine learning techniques." *International Journal of Computer Applications*, 182(47), 35–40.

[7] Liu, Y., Reiter, M. K., & Ning, P. (2011). "False data injection attacks against state estimation in electric power grids." *ACM Transactions on Information and System Security*, 14(1), 13

[8] Khalaf, B. A., & Bakar, K. A. (2020). "A comprehensive study of machine learning-based SCADA intrusion detection systems." *IEEE Access*, 8, 84654–84672.

[9] Choudhary, S., & Soni, D. (2018). "SCADA system security: Threats and protective measures." *International Journal of Advanced Research in Computer Science*, 9(2), 167–172.

[10] Moustafa, N., & Slay, J. (2015). "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6.