TinyBasic Plus on Arduino Nano

S.Y.B.sc(CS)

Mitesh S. Gorad

PUNE DISTRICT EDUCATION ASSOCIATION

PROF.RAMKRISHNA MORE
ARTS, COMMERCE & SCIENCE COLLEGE

AKURDI, PRADHIKARAN PUNE -411044.

DEPARTMENT OF ELECTRONICS

CERTIFICATE

This is to certify that,

Mr. Gorad Mitesh Sudhakar of class **S. Y. B.Sc. (CS)** Div. **B** Roll No. **41449** has satisfactorily completed necessary project work in ELECTRONICS during the academic year 2023-2024.

Exam seat no: _____

Practical in charge:

Date: / /

H.O.D Dept. of Electronics	
Internal Examinar	
External Examinar	

INDEX

1. INTRODUCTION:

- Imagine a small, but powerful computer that you can program easily. Well, that's what happens when you combine TinyBASIC Plus (TBXL) with an Arduino Nano!

- TBXL is a simple programming language created back in the 1970s by Li-Chen Wang. It's like the LEGO of coding - easy to understand and use. On the other hand, the Arduino Nano is a tiny but mighty computer board that can do lots of cool stuff.

- By putting TBXL onto the Arduino Nano, we've created a basic 8/16 bit computer that's perfect for learning and experimenting. You can write simple programs to make lights blink, play sounds, or even control robots!

- This combination of old-school simplicity with modern technology opens up a world of possibilities. Whether you're a curious beginner or a seasoned tech enthusiast, the TBXL-Arduino Nano computer is a fun way to explore the basics of programming and electronics.

2. COMPONENTS & COST:

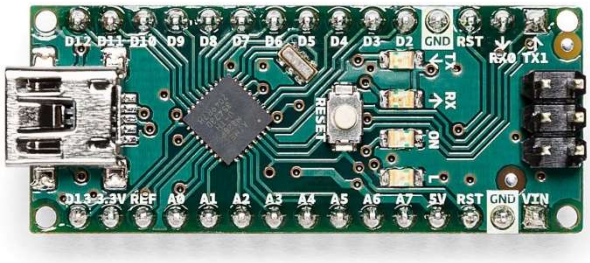
SR. NO.	COMPONENT NAME	QUANTITY	PRICE
1	Arduino Nano	2	2x350=700
2	VGA/DB-15 Port	1	30
3	PS2/MINI-DIN-6 Port	1	30
4	1x40 pin Female Header Pin	3	3x20=60
5	1x40 Pin Male Header Pin	1	20
6	Single Core Wire	3-M	3x10=30
7	Resistor (68ohm,470ohm,1kohm)	3	3x5=15
8	Double Sided Green PCB (9x15CM)	1	150
9	Double Sided Green PCB (2x8CM)	1	50
10	4 Pin Push Button	2	2x10=20
12	LED Diode (Red & Green)	1+1	2x5=10
TOTAL			1,115/-

So. No.	Name of Tools
1	Soldering Iron
2	Solder Wire
3	PS2-Keyboard

3. INFORMATION OF COMPONENTS:

1. Arduino Nano:

The Arduino Nano is a small, compact 8-bit microcontroller board based on the ATmega328P microcontroller, offering similar functionalities to the Arduino Uno in a smaller form factor. Here's a detailed overview:

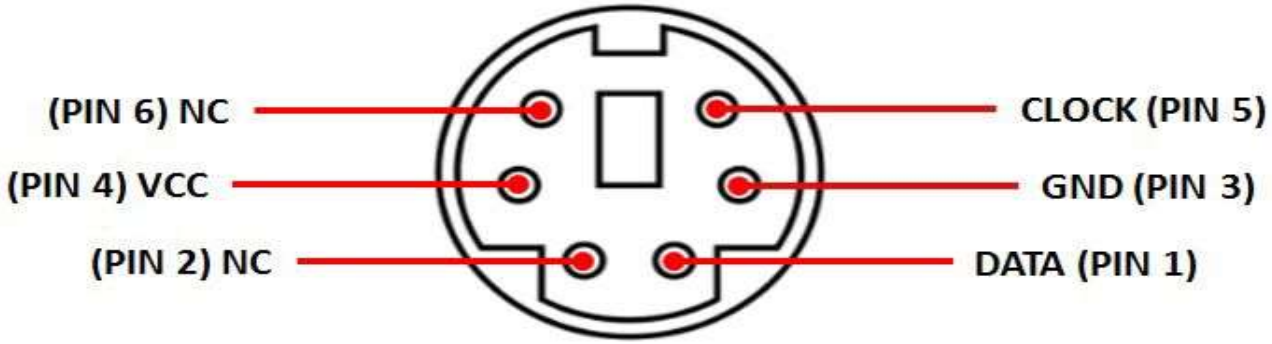


- **Arduino Nano:**
  a. Microcontroller: ATmega328P
  b. Operating Voltage: 5V
  c. Input Voltage (recommended): 7-12V
  d. Input Voltage (limits): 6-20V
  e. Digital I/O Pins: 14 (of which 6 provide PWM (Pulse Width Modulation) output)
  f. Analog Input Pins: 8
  g. DC Current per I/O Pin: 40 mA
  h. DC Current for 3.3V Pin: 50 mA
  i. Flash Memory: 32 KB (ATmega328P) of which 2 KB used by bootloader
  j. SRAM: 2 KB (ATmega328P)
  k. EEPROM: 1 KB (ATmega328P)
  l. Clock Speed: 16 MHz
  m. LED_BUILTIN: Pin 13
  n. Length: 45 mm
  o. Width: 18 mm
  p. Weight: 5 g

- **Features:**
  a. Compact Size: The Nano is incredibly small, making it suitable for projects where space is limited.
  b. Easy to Use: It can be programmed with the Arduino IDE (Integrated Development Environment), making it accessible to beginners and experienced users alike.
  c. USB Interface: It features a built-in USB interface for programming and serial communication, eliminating the need for additional hardware.
  d. Power Options: It can be powered via USB connection or an external power supply, providing flexibility in different project scenarios.
  e. Variety of I/O Pins: Despite its small size, it offers a generous number of digital and analog I/O pins, allowing for versatile project designs.
  f. Compatible with Shields: It is compatible with most Arduino shields, enabling users to extend its capabilities easily.
  g. Integrated Voltage Regulator: The onboard voltage regulator ensures stable operation even when powered by a wide range of input voltages.
  h. Integrated LEDs: It includes built-in LEDs for power and status indication, aiding in troubleshooting and debugging.
  i. Cost-Effective: Despite its features, the Nano remains cost-effective, making it a popular choice for hobbyists and professionals alike.

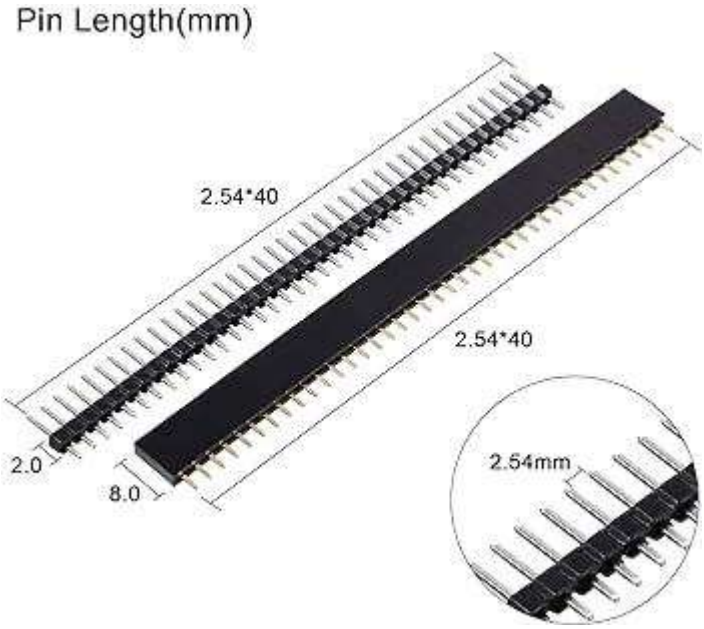
- **Applications:**
  a. Prototyping: Ideal for rapid prototyping due to its small size and ease of use.
  b. Embedded Systems: Suitable for embedded systems development, particularly in applications where space is limited.
  c. Education: Widely used in educational settings to teach programming, electronics, and robotics.
  d. DIY Projects: Popular among hobbyists for creating various DIY projects, from simple LED displays to complex robotics.
  e. IoT Devices: Used in developing Internet of Things (IoT) devices and projects due to its compact size and versatility.

2. PS2/MINI-DIN-6 Port:



The PS2/MINI-DIN-6 port, commonly known as the PS/2 port, is a connector for keyboards and mice. Introduced by IBM, it features a round shape with six pins arranged in a circular pattern. The port provides reliable, bidirectional data transmission between peripherals and computers, known for its stability compared to USB. It supports hot-swapping, allowing peripherals to be connected or disconnected without rebooting. Found in desktops and industrial systems, the PS2 port maintains legacy support despite USB's prevalence. Its simple design and dedicated functionality make it a trusted interface for user input in various computing environments.

3. 1x40 pin F&M Header Pins:

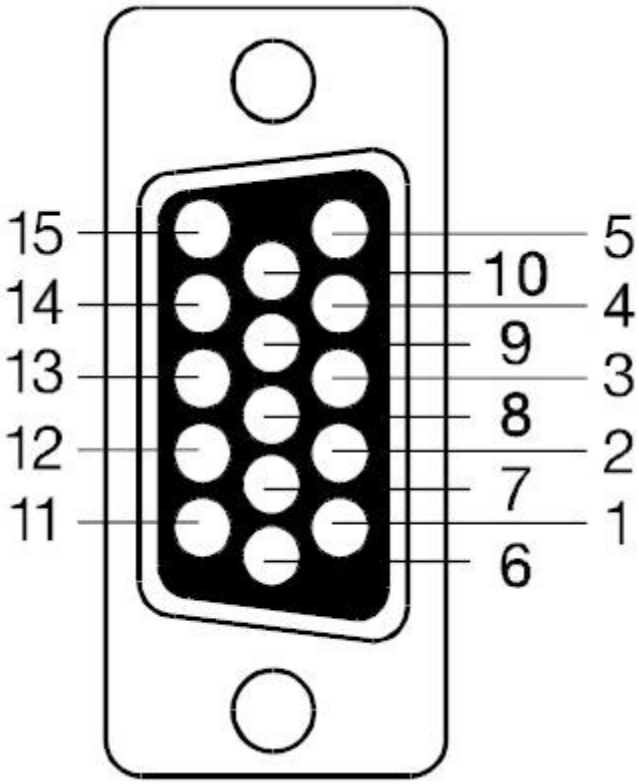


1x40 pin male and female header pins are essential electronic components used for creating connections between electronic modules, boards, and devices. The male header pins consist of metal pins protruding from a plastic housing, while the female header pins feature plastic housings with receptacles or sockets for accepting the male pins. These headers are typically used in pairs, with the male pins soldered onto one PCB (Printed Circuit Board) and the female headers soldered onto another, allowing for easy connection and disconnection between the two boards. Male header pins are versatile and commonly used on circuit boards, modules, and development platforms to provide external connection points. They facilitate easy interfacing with other electronic components or devices. Female header pins, on the other hand, serve as mating connectors for male header pins, providing secure connections between boards.

These header pins are available in various lengths, pin spacings, and configurations to accommodate different applications and requirements. They are widely used in prototyping, DIY electronics, and circuit design due to their versatility, ease of use, and reliability in creating electrical connections.

4. **VGA/DB-15 Port:**

The VGA/DB15 pin, also known as the Video Graphics Array connector, is a standard interface used for transmitting analog video signals between computers or video sources and display devices such as monitors, projectors, or TVs. The connector features a trapezoidal shape with 15 pins arranged in three rows. It supports the transmission of analog video signals, including red, green, blue (RGB) color components, horizontal and vertical synchronization signals, and ground connections.



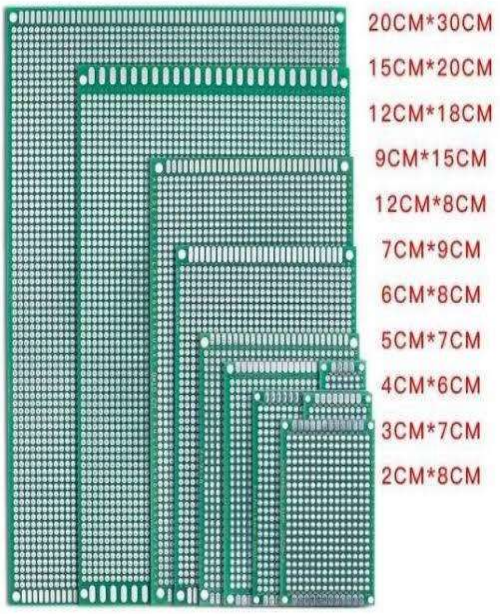
VGA ports are commonly found on computers, laptops, and graphics cards, providing video output for desktop PCs and workstations. They are also present on projectors, allowing them to be connected to computers or other video sources for presentations and multimedia playback.

Despite the emergence of digital video interfaces like HDMI and DisplayPort, VGA ports continue to be used in many applications, especially in legacy systems, industrial equipment, and older display devices lacking digital inputs. VGA supports a wide range of resolutions and refresh rates, making it a versatile connector for various display requirements. However, its analog nature may result in degraded image quality at higher resolutions and longer cable lengths compared to digital interfaces.

5. **Single Core Wire:**

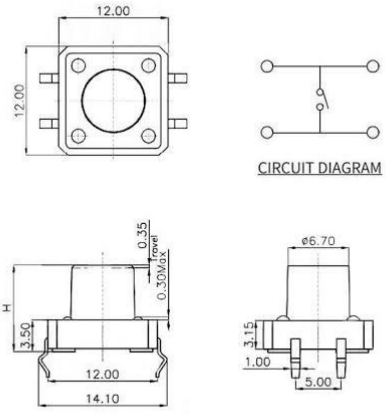
Single core wire consists of a single strand of conductive material, typically copper or aluminum, surrounded by insulation. It is commonly used in electrical wiring for various applications such as power distribution, lighting, and electronics. Single core wire is flexible and easy to work with, making it suitable for both residential and commercial installations. It comes in different gauges or thicknesses to accommodate different current-carrying capacities and voltage ratings. Single core wire is available in various insulation materials, including PVC, rubber, and silicone, providing options for different environmental conditions and applications.

6. **Double Sided Green PCB:**



A Double-Sided Green PCB (Printed Circuit Board) is a type of circuit board with conductive traces and components mounted on both sides. The green color typically comes from a solder mask applied to the surface of the PCB for insulation and protection. These PCBs are versatile and widely used in various electronic applications due to their increased circuit complexity and component density compared to single-sided boards. Conductive vias or holes are used to connect traces on different layers, allowing for the creation of complex circuits with connections between components on both sides of the board. Double-sided PCBs are cost-effective for many applications, offering increased functionality and flexibility without significantly higher manufacturing costs. Designing double-sided PCBs requires careful planning to ensure proper routing of traces, component placement, and signal integrity. They are used in consumer electronics, industrial equipment, automotive systems, medical devices, communication equipment, and power supplies. Overall, double-sided green PCBs offer a balance of complexity, cost-effectiveness, and versatility, making them suitable for a wide range of electronic applications.

7. **4-pin Push Button:**



A 4-pin push button PCB (Printed Circuit Board) typically consists of four electrical contacts arranged in a specific configuration. These push buttons are commonly used in electronic devices for various functions such as power on/off, mode selection, or menu navigation. The four pins serve specific purposes: two pins are for the actual switch mechanism, while the other two are for providing stability and secure mounting on the PCB.

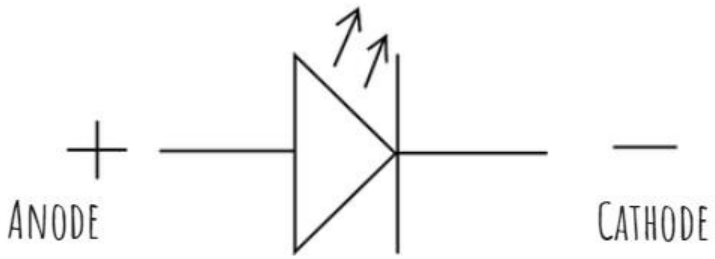
The push button's design allows it to complete or break an electrical circuit when pressed, triggering the desired action in the device it's integrated with. PCB-mounted push buttons are advantageous for their compact size, ease of installation during PCB assembly, and reliability in providing tactile feedback to users.

In terms of construction, the push button typically comprises a housing, a plunger, and internal contacts. When the plunger is depressed, it makes contact with the internal mechanism, completing the circuit and allowing current to flow. Once released, the spring-loaded plunger returns to its original position, breaking the circuit.

These buttons are available in various sizes, shapes, and actuation forces to suit different applications and user preferences. Common variants include momentary and latching types, offering either

temporary or sustained activation, respectively. Overall, 4-pin push button PCBs are fundamental components in electronic design, offering versatility, reliability, and ease of integration.

8. **LED Diode:**



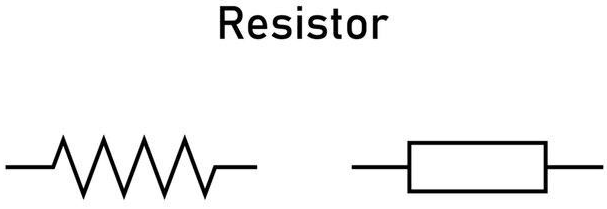
LED diodes, specifically red and green variants, are semiconductor devices that emit light when current flows through them. Red LEDs typically use a gallium arsenide phosphide (GaAsP) semiconductor material, while green LEDs typically use a combination of gallium nitride (GaN) and indium gallium nitride (InGaN).

In operation, when voltage is applied to the LED, electrons and holes recombine within the semiconductor material, releasing energy in the form of photons. The color of light emitted depends on the energy band gap of the semiconductor material, with red LEDs emitting longer wavelengths (~620-750 nanometers) and green LEDs emitting shorter wavelengths (~490-570 nanometers).

Red LEDs are commonly used in applications such as indicators, brake lights, and displays, while green LEDs find applications in traffic lights, digital clocks, and instrumentation displays. Both red and green LEDs are preferred for their energy efficiency, durability, and compact size compared to traditional light sources.

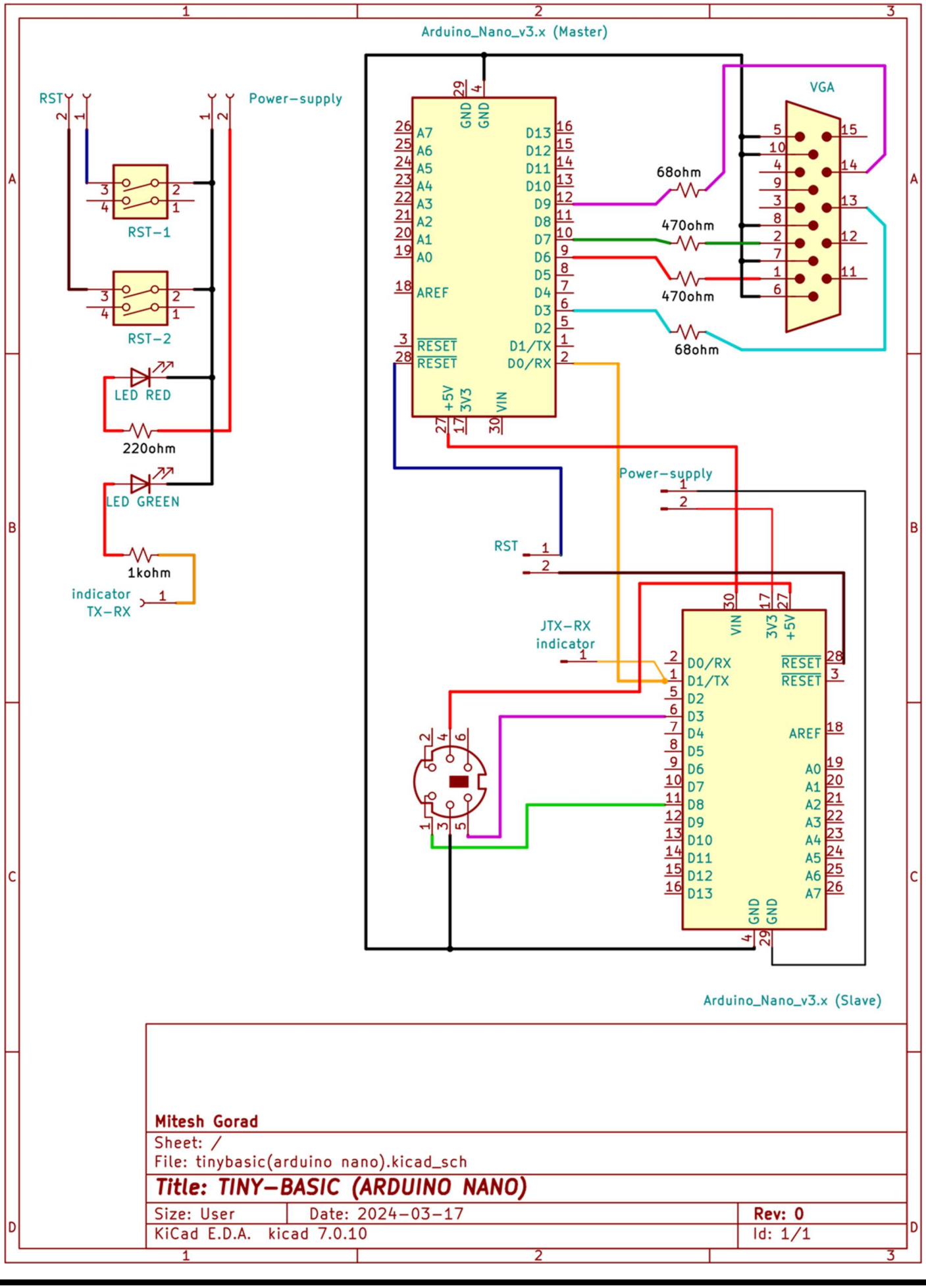
LED diodes offer several advantages, including low power consumption, long lifespan, instant illumination, and resistance to shock and vibration. They are also environmentally friendly due to their lack of hazardous materials such as mercury. As a result, red and green LED diodes have become ubiquitous in various lighting and display applications, contributing to energy efficiency and sustainability efforts.

9. **Resistor:**



Resistors are electrical components that impede the flow of current in a circuit, measured in ohms (Ω). They come in various types like carbon film, metal film, and wire wound, each with specific characteristics. Resistors are crucial for controlling voltage levels, limiting current, and dividing voltages in electronic circuits. They follow Ohm's law, where the current passing through a resistor is directly proportional to the voltage applied and inversely proportional to its resistance. Resistors dissipate energy as heat, so they have power ratings indicating their maximum power handling capacity. Overall, resistors play a vital role in circuit design, ensuring proper functionality and protecting electronic components from damage due to excessive current flow.

4. CIRCUIT DIAGRAM:



5. Working of Diagram:

This project implements a basic computer system using two Arduino Nano boards: one acting as the master controller connected to a VGA display, and the other serving as a slave controller connected to a PS2 keyboard port. The master Nano controls the VGA output for visual display, while the slave Nano manages input from the PS2 keyboard. Additionally, the slave Nano provides power and ground connections to the master Nano. The master and slave communicate via serial communication, enabling coordination between input and output functionalities.

1. Master Arduino Nano (Connected to VGA):

- The master Nano serves as the primary controller, responsible for driving the VGA display.
- Data pins D9, D7, D6, and D3 of the Arduino Nano are connected to 14,2,1, and 13 pins on the VGA connector to send video signals.
- Pins 5, 6, 7, 8, and 10 of the VGA connector are connected to ground (GND) pins of the slave Nano and also to the master Nano's ground pin to establish a common ground reference.
- The master Nano receives power (5V) from the slave Nano's VIN pin to ensure consistent power supply.

Master Arduino Nano	VGA (Video Graphics Array)
D9	14 (V-Sync)
D7	2 (Green)
D6	1 (Red)
D3	13 (H-Sync)

Slave Arduino Nano	VGA (Video Graphics Array)
GND	5,6,7,8,10

2. Slave Arduino Nano (Connected to PS2 Port):

- The slave Nano functions as an input device controller, connected to a PS2 keyboard port.
- Pin D3 of the Nano is connected to the clock (CLK) pin 5 of the PS2 port to receive clock signals.
- Pin D8 of the Nano is connected to the data pin (DATA) 1 of the PS2 port to receive data signals.
- Pins 4 and 3 of the PS2 port are connected to 5V and GND respectively to provide power and ground connections.

Slave Arduino Nano	PS2
D3	1
D8	5
GND	3
+5V	4

3. Master-Slave Communication:

- The master Nano communicates with the slave Nano via serial communication.
- The master's RX pin is connected to the slave's TX pin, allowing the master to send commands or data to the slave.
- This communication link facilitates coordination between the master (handling VGA output) and the slave (handling PS2 input).

Master Arduino Nano	Slave Arduino Nano
RX	TX

4. Overall Functionality:
The master Nano controls the VGA display to output visual information, while the slave Nano manages input from a PS2 keyboard and provides power and ground connections to the master Nano. Through serial communication, the master and slave coordinate their tasks, creating a basic computer system using Arduino Nano boards.

5. Future Considerations:
Potential enhancements include incorporating additional peripherals, expanding input/output capabilities, and implementing more advanced functionalities to enhance the overall capabilities of the system.

6. About Tiny-Basic:

Tiny BASIC, a minimalist version of the BASIC programming language, has a rich history spanning from its inception in the 1970s to its continued influence on programming education and retro computing enthusiasts today.

1) Origins and Early Development (1970s):

1. The roots of Tiny BASIC can be traced back to the work of John Kemeny and Thomas Kurtz at Dartmouth College in the early 1960s, who created the BASIC programming language to make computer programming accessible to novices.
2. In 1975, Dr. Li-Chen Wang developed the first version of Tiny BASIC for the Intel 8080 microprocessor. This version was designed to run on early microcomputers with limited memory and processing power, such as the Altair 8800.
3. Tiny BASIC aimed to retain the simplicity and ease of use of the original BASIC language while reducing its memory footprint to fit within the constraints of microcomputers of the time.

2) Evolution and Popularization (Late 1970s-Early 1980s):

1. Tiny BASIC quickly gained popularity among hobbyists and early computer enthusiasts due to its simplicity and suitability for low-resource systems.
2. Various versions and adaptations of Tiny BASIC were developed and ported to different microcomputer platforms, including the Apple II, TRS-80, and Commodore PET, among others.
3. The publication of Dr. Li-Chen Wang's article in the "Dr. Dobb's Journal" in 1976, detailing the implementation of Tiny BASIC, further fueled its adoption and dissemination within the hobbyist community.

3) Legal Challenges and Variants (1980s-1990s):

1. The widespread adoption of Tiny BASIC and its variants led to legal disputes over copyright and intellectual property rights.
2. Microsoft's development and dominance of the BASIC language for early personal computers, known as Microsoft BASIC, overshadowed Tiny BASIC and other variants in the commercial market.
3. Despite legal challenges, Tiny BASIC continued to be developed and modified by enthusiasts, resulting in the creation of numerous variants and adaptations tailored to specific hardware platforms and user preferences.

4) **Legacy and Modern Revival (2000s-Present):**
   1. While the popularity of Tiny BASIC waned in the commercial market by the late 1980s and 1990s, its legacy endured in the realm of programming education and retro computing.
   2. The simplicity and accessibility of Tiny BASIC make it an ideal tool for teaching programming concepts to beginners and students.
   3. Today, Tiny BASIC interpreters and emulators are available for various platforms, allowing enthusiasts to experience the early days of microcomputing and programming firsthand.

7. About Tiny-Basic Plus & C-implementation of it:

Tiny BASIC Plus is an enhanced version of the Tiny BASIC programming language, which was originally developed by Dr. Li-Chen Wang in the 1970s. Tiny BASIC Plus expands upon the functionality of the original Tiny BASIC by adding features like string handling, improved error messages, and additional commands. It was designed to be easy to understand and implement on small computers with limited resources.

As for its implementation in C, there are several versions of Tiny BASIC Plus implemented in C available online, created by various developers. These implementations aim to preserve the simplicity and functionality of the original Tiny BASIC Plus while leveraging the flexibility and power of the C programming language.

Unfortunately, specific dates for the development of Tiny BASIC Plus and its implementations in C may vary depending on the specific version and developer. However, the original Tiny BASIC was developed in the mid-1970s, so Tiny BASIC Plus and its C implementations likely emerged sometime after that, continuing into the following decades as hobbyists and developers adapted the language to different computing environments.

8. Supported Statements and Functions TBXL-Arduino:

- **System**
  - BYE - *exits Basic, soft reboot on Arduino*
  - END - *stops execution from the program, also "STOP"*
  - MEM - *displays memory usage statistics*
  - NEW - *clears the current program*
  - RUN - *executes the current program*

- **File IO/SD Card**
  - FILES - *lists the files on the SD card*
  - LOAD filename.bas - *loads a file from the SD card*
  - CHAIN filename.bas - *equivalent of: new, load filename.bas, run*
  - SAVE filename.bas - *saves the current program to the SD card, overwriting*

- **EEProm - nonvolatile on-chip storage**
  - EFORMAT - *clears the EEProm memory*
  - ELOAD - *load the program in from EEProm*
  - ESAVE - *save the current program to the EEProm*
  - ELIST - *print out the contents of EEProm*
  - ECHAIN - *load the program from EEProm and run it*

- **IO, Documentation**
  - INPUT variable - *let the user input an expression (number or variable name)*
  - PEEK (address) - *get a value in memory* (unimplemented)
  - POKE address - *set a value in memory* (unimplemented)
  - PRINT expression - *print out the expression, also "?"*
  - REM stuff - *remark/comment, also " ' "*

- **Expressions, Math**
  - A=V, LET A=V - *assign value to a variable*
  - +, -, *, / - *Math*
  - <,<=,=,<>,!=,>=,> - *Comparisons*
  - ABS(expression) - *returns the absolute value of the expression*
  - RSEED(v) - *sets the random seed to v*
  - RND(m) - *returns a random number from 0 to m*

- **Control**
  - IF expression statement - *perform statement if expression is true*
  - FOR variable = start TO end - *start for block*
  - FOR variable = start TO end STEP value - *start for block with step*
  - NEXT - *end of for block*
  - GOTO linenumber - *continue execution at this line number*
  - GOSUB linenumber - *call a subroutine at this line number*
  - RETURN - *return from a subroutine*

- **Pin IO**
  - DELAY timems*- wait (in milliseconds)*
  - DWRITE pin,value - *set pin with a value (HIGH,HI,LOW,LO)*
  - AWRITE pin,value - *set pin with analog value (pwm) 0..255*
  - DREAD(pin) - *get the value of the pin*
  - AREAD(analogPin) - *get the value of the analog pin*

    NOTE: "PINMODE" command removed as of version 0.11

- **Sound - Piezo wired with red/+ on pin 5 and black/- to ground**
  - TONE freq,timems - play "freq" for "timems" milleseconds (1000 = 1 second)
  - TONEW freq,timems - same as above, but also waits for it to finish
  - NOTONE - stop playback of all playing tones

    NOTE: TONE commands are by default disabled

9. Example programs:

Here are a few example programs to get we started...

- Try

  Let's print 'Hello, world.'

```
10 PRINT ”Hello, World”
20 GOTO 10
RUN
```

- User Input

  Let a user enter a new value for a variable, enter a number like '33' or '42', or a varaible like 'b'.

```
10 A=0
15 B=999
20 PRINT "A is ", A
30 PRINT "Enter a new value ";
40 INPUT A
50 PRINT "A is now ", A
```

- **Blink**

  hook up an LED between pin 3 and ground

```
10 FOR A=0 TO 10
20 DWRITE 3, HIGH
30 DELAY 250
40 DWRITE 3, LOW
50 DELAY 250
60 NEXT A
```

- **Fade**

  hook up an LED between pin 3 and ground

```
10 FOR A=0 TO 10
20 FOR B=0 TO 255
30 AWRITE 3, B
40 DELAY 10
50 NEXT B
60 FOR B=255 TO 0 STEP -1
70 AWRITE 3, B
80 DELAY 10
90 NEXT B
100 NEXT A
```

- **LED KNOB**

  hook up a potentiometeter between analog 2 and ground, led from digital 3 and ground. If knob is at 0, it stops

```
10 A = AREAD( 2 )
20 PRINT A
30 B = A / 4
40 AWRITE 3, B
50 IF A == 0 GOTO 100
60 GOTO 10
100 PRINT "Done."
```

- **ECHAIN example**

Write a small program, store it in EEPROM. We'll show that variables don't get erased when chaining happens

```
EFORMAT
10 A = A + 2
20 PRINT A
30 PRINT "From eeprom!"
40 IF A = 12 GOTO 100
50 PRINT "Shouldn't be here."
60 END
100 PRINT "Hi!"
```

Then store it in EEProm

```
ESAVE
```

Now, create a new program in main memory and run it

```
NEW
10 A = 10
20 PRINT A
30 PRINT "From RAM!"
40 ECHAIN
```

List both, and run

```
ELIST
LIST
RUN
```

10.Code of The Circuit:

Sr. No.	Arduino Nano	OR-Code
1	**Master**	
2	**Slave**	

THANK YOU