

# Scientific Calculator

EE24BTECH11066 - YERRA AKHILESH

## 1 CONNECTIONS

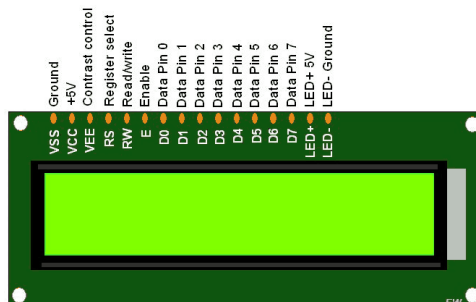
### 1.1 Connecting the LCD Display (Left to Right Order)

The 16x2 LCD is interfaced with the Arduino using the 4-bit mode connection as follows:

LCD Pin	Connected to Arduino Pin
VSS	GND
VDD	5V
VO	Middle Pin of Potentiometer
RS	Digital Pin 1
RW	GND
E	Digital Pin 0
D0	(Not Connected)
D1	(Not Connected)
D2	(Not Connected)
D3	(Not Connected)
D4	Digital Pin 5
D5	Digital Pin 4
D6	Digital Pin 3
D7	Digital Pin 2
A (LED+)	5V
K (LED-)	GND

### Potentiometer Wiring:

- Leftmost Pin → GND
- Middle Pin → LCD VO (Contrast Pin)
- Rightmost Pin → 5V



1.2 Connecting the Input Buttons

Each button is connected in pull-up mode, meaning:

- One terminal of each button goes to an Arduino input pin.
- The other terminal is connected to GND.

Button Function	Connected to Arduino Pin
Number 0	Digital Pin 13
Number 1	Digital Pin 12
Number 2	Digital Pin 11
Number 3	Digital Pin 10
Number 4	Digital Pin 9
Number 5	Digital Pin 8
Number 6	Digital Pin 7
Number 7	Digital Pin 6
Number 8	Analog Pin A0
Number 9	Analog Pin A1
Operator Selection	Analog Pin A2
Trigonometric Function	Analog Pin A3
Reset Button	Analog Pin A4
Enter (Evaluate Expression)	Analog Pin A5

- Each button’s other pin is connected to GND

1.3 Compiling and Uploading the Code

1.3.1 Compilation:

```
avr-gcc -mmcu=atmega328p -DF_CPU=16000000UL -Os -o scientific.elf scientific.c
avr-objcopy -O ihex scientific.elf scientific.hex
```

1.3.2 Uploading part:

```
sudo avrdude -c arduino -P /dev/ttyACM0 -b 115200 -p atmega328p -U flash:w:scientific.hex
```

Ensure that the microcontroller is connected to the correct port (/dev/ttyACM0) before executing the upload command.

2 MAIN FEATURES

- **LCD Display (4-bit Mode):** Uses a 4-bit connection to show numbers and results.
- **14 Buttons for Input:** Includes 10 number buttons (0-9), a decimal point button, an operator selection button, a trigonometric function button, and an enter button.
- **Basic Math Operations:** Supports addition (+), subtraction (−), multiplication (×), and division (÷).

- **Trigonometric Functions:** Implements  $\sin$ ,  $\cos$ ,  $\tan$ , and their inverse functions ( $\sin^{-1}$ ,  $\cos^{-1}$ ,  $\tan^{-1}$ ).
- **Angle in Degrees:** Trigonometric calculations use degrees instead of radians.
- **Handles Long Calculations:** Supports multiple steps and follows operator precedence.
- **Error Detection:** Detects mistakes like division by zero and prevents incorrect calculations.

### 3 HOW BUTTONS WORK

#### 3.1 Operator Button

The arithmetic operator button cycles through different operations with each press:

- 1) First press: Addition (+)
- 2) Second press: Subtraction (−)
- 3) Third press: Multiplication (×)
- 4) Fourth press: Division (÷)
- 5) Fifth press: Returns to addition (+)

The LCD displays the selected operator so the user knows the current choice.

#### 3.2 Trigonometric Function Button

This button cycles through trigonometric functions in the following order:

- 1) First press:  $\sin$
- 2) Second press:  $\cos$
- 3) Third press:  $\tan$
- 4) Fourth press:  $\sin^{-1}$
- 5) Fifth press:  $\cos^{-1}$
- 6) Sixth press:  $\tan^{-1}$
- 7) Seventh press: Returns to  $\sin$

The LCD displays the chosen function, allowing the user to input a value.

### 4 HOW CODE WORKS

#### 4.1 Math Calculations

- Uses Taylor series for trigonometric calculations.
- Does not need extra math libraries.
- Checks if values for inverse trigonometric functions are valid.

#### 4.2 User Interface

- Single button selection for arithmetic and trigonometric functions.
- Button debouncing prevents multiple accidental inputs.
- The LCD clearly displays the current equation.
- Results are displayed with appropriate precision.

### 4.3 Expression Calculation

- Correctly handles brackets and follows standard mathematical rules.
- Uses recursive parsing to evaluate complex expressions.

### 4.4 Memory Usage

- Efficient storage for microcontroller use.
- Remembers the last result for continuous calculations.

## 5 SPECIAL FEATURES

- Supports floating-point calculations.
- Handles multi-step expressions correctly.
- Portable design for use with different hardware.
- Optimized for low-memory usage.

## 6 EASE OF USE

- Results can be used in further calculations.
- Single-button cycling makes operations simple.
- Prevents invalid inputs like multiple decimal points.
- Automatically fixes missing brackets in expressions.

## 7 FINAL SUMMARY

This calculator is designed to be simple, efficient, and capable of handling complex calculations. It effectively manages memory and provides a smooth user experience, making it a great example of embedded system programming and mathematical implementation.

