

Eta32mini

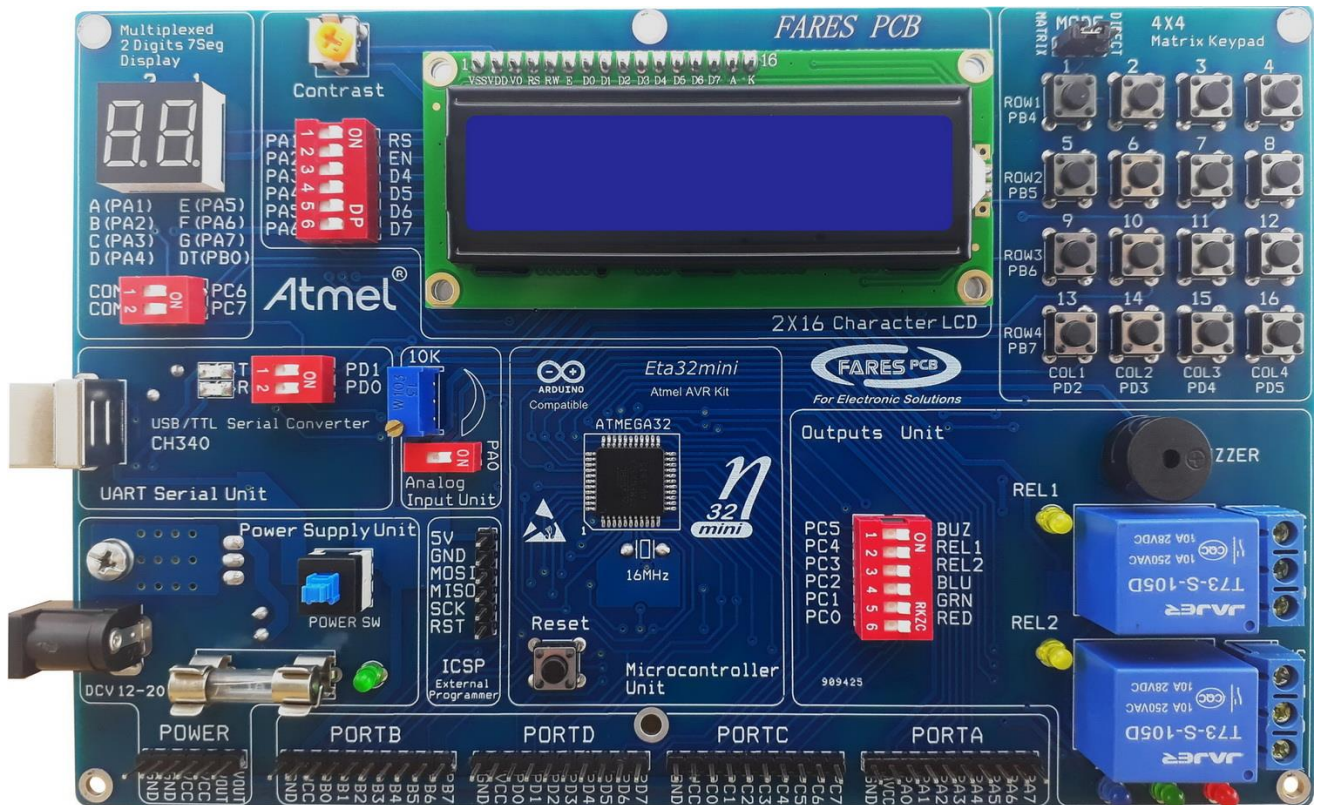
ATMEL AVR Development Kit

General Description

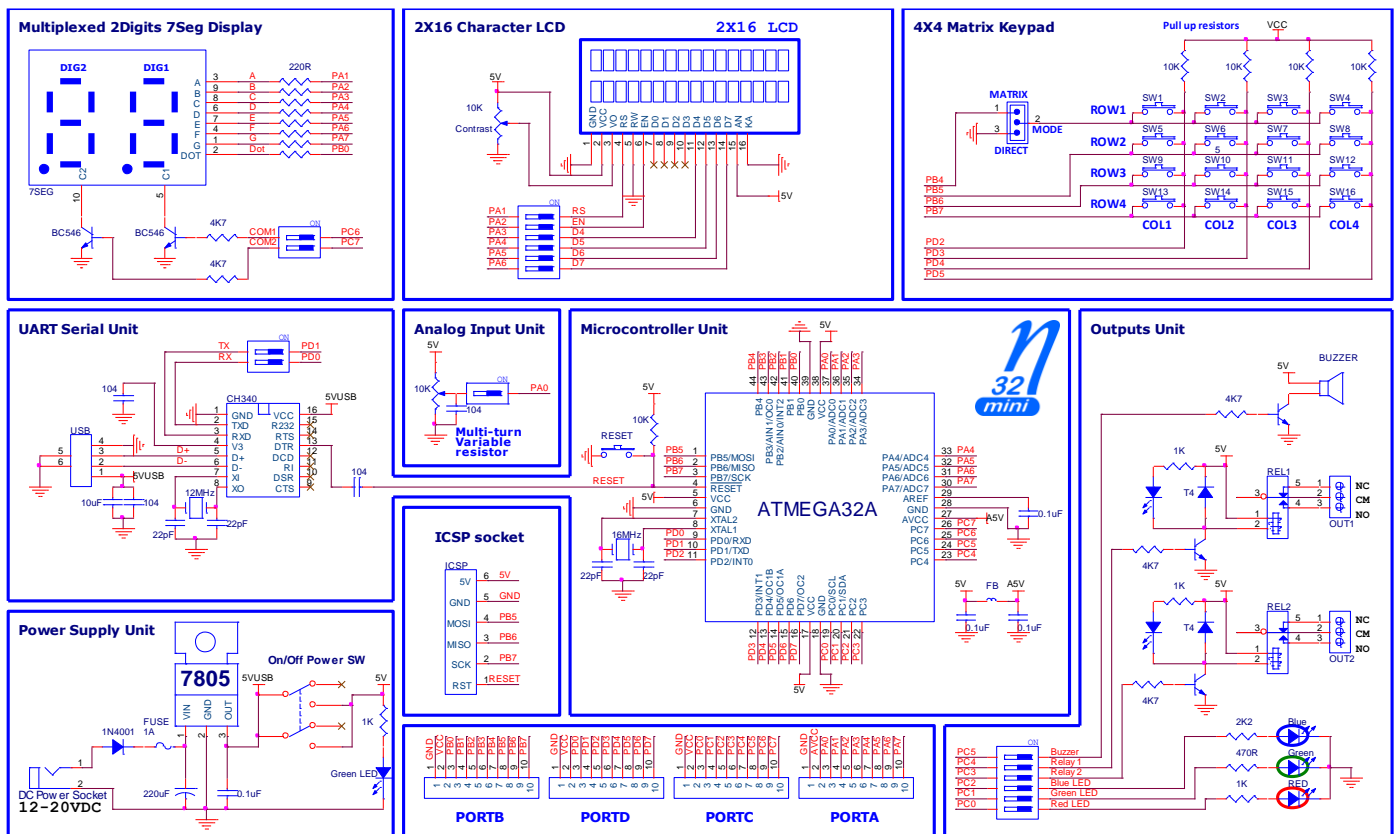
Eta32mini kit is an AVR development board designed particularly for students, beginners and recent graduated engineers to provide easy developing of AVR microcontroller projects.

Eta32mini supports ATMEGA32 microcontroller chip. *Eta32mini* kit provides the most common primary devices and circuits, such as LCD, KEYPAD, serial interface, and more. Thus, it saves most of the wasted time and efforts in hardware design and test, hence, a developer can focus his efforts on firmware development. All I/O pins are brought out via pin header for direct port accessing.

Eta32mini kit comes preprogrammed with UART bootloader, which eliminates the need for external programmer (programming using Eta32 burner from FARESPCB Co). However, kit can be programmed using Arduino IDE (another bootloader). Also standard 6 pin header socket is included (ICSP) for external programmers.



***Eta32mini* kit**



***Eta32mini* kit (schematic)**

***Eta32mini* key features**

- Power supply unit.
- ATMEGA32 Microcontroller with reset switch and crystal oscillator.
- 4X4 matrix keypad.
- Output unit involves three output LEDs, two output relays and one output buzzer.
- Multiplexed two digits 7 segments display.
- 2X16 LCD display.
- Variable analog input using high precise multi-turn resistor.
- Virtual COM port via USB.
- Standard 6 pin header socket ICSP for external programmers.
- All microcontroller I/O pins are brought out via pin header.

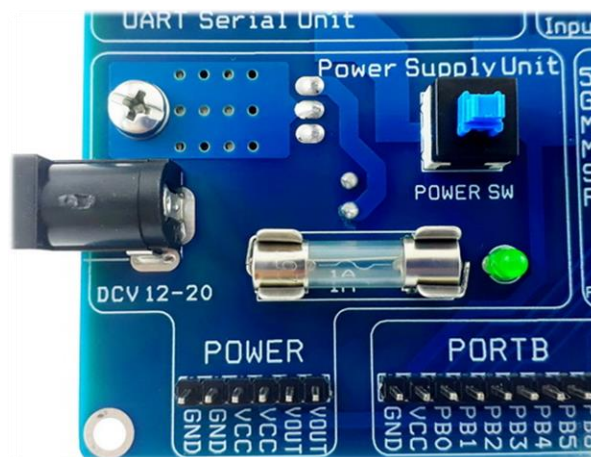
Power Supply Unit

Power could be supplied from DC power supply adaptor via DC power socket (12V - 20V) or from USB power via USB type B socket.

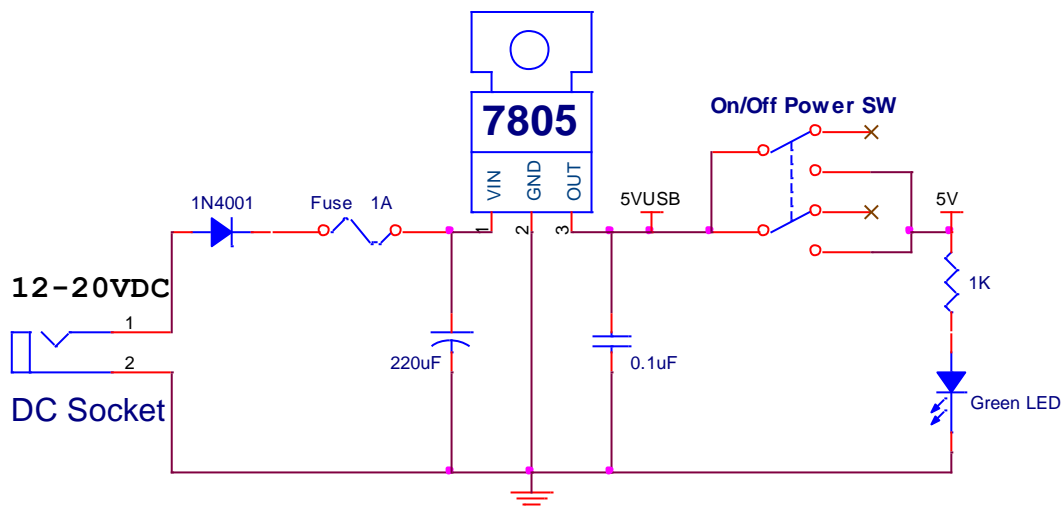
Power supply unit features:

- On/Off Power switch.
- Green LED power indicator.
- 7805 (5V power regulator 5% tolerance).
- 1A Fuse for over current protection.
- GND , VCC(5V) and adaptor voltage(VOUT) are brought out for external using via header sockets.

Note: *Eta32mini* kit is protected against reversed polarity of power.



Power supply unit



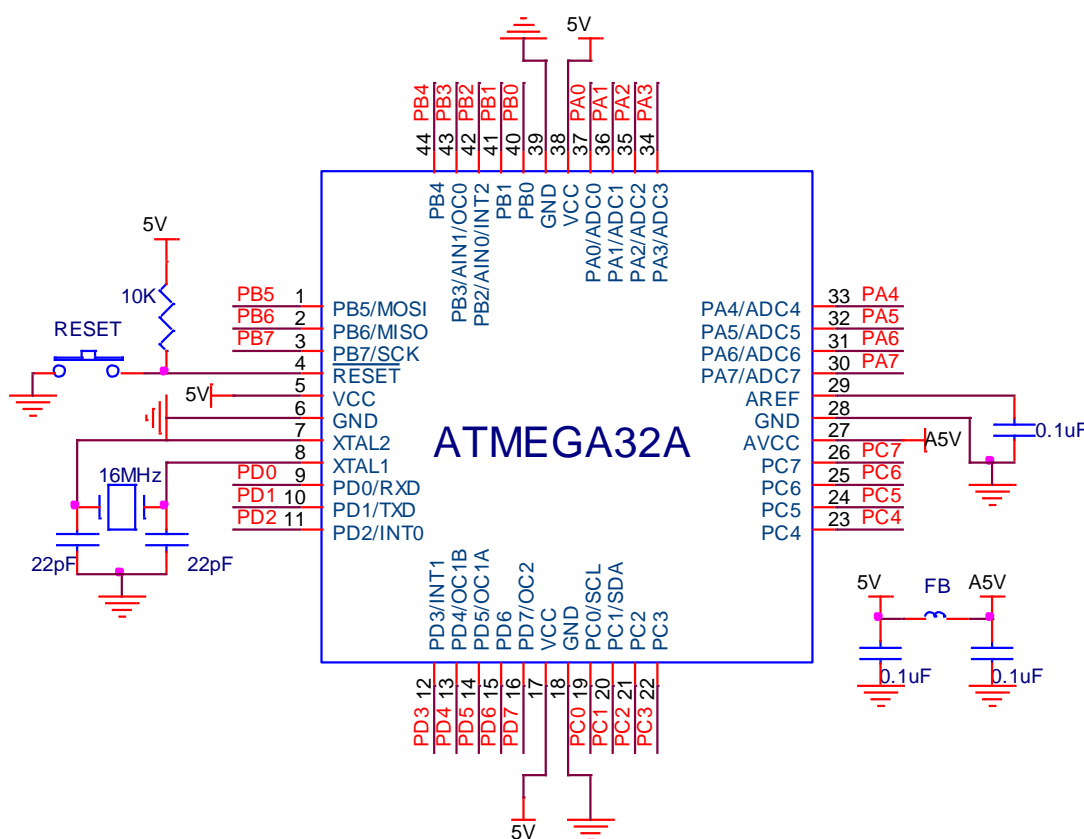
Power supply unit (schematic)

Microcontroller Unit

44 pin TQFP ATMEGA32A chip with 16 MHz crystal oscillator and Push button reset switch.



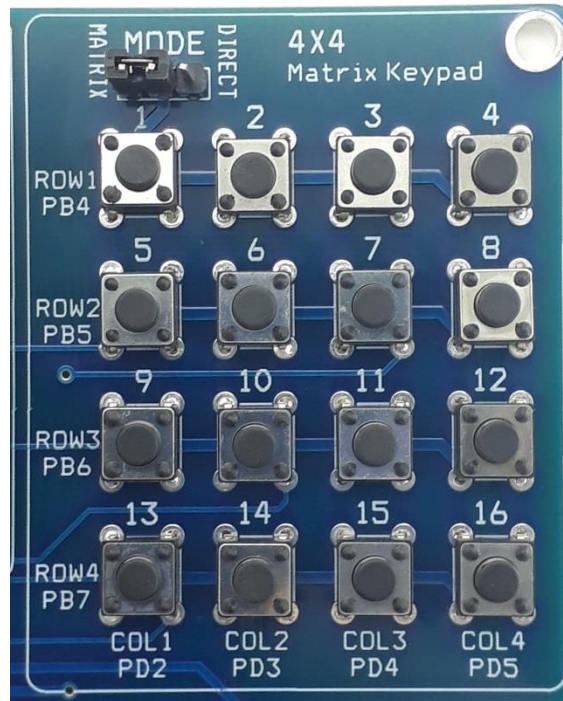
Microcontroller unit



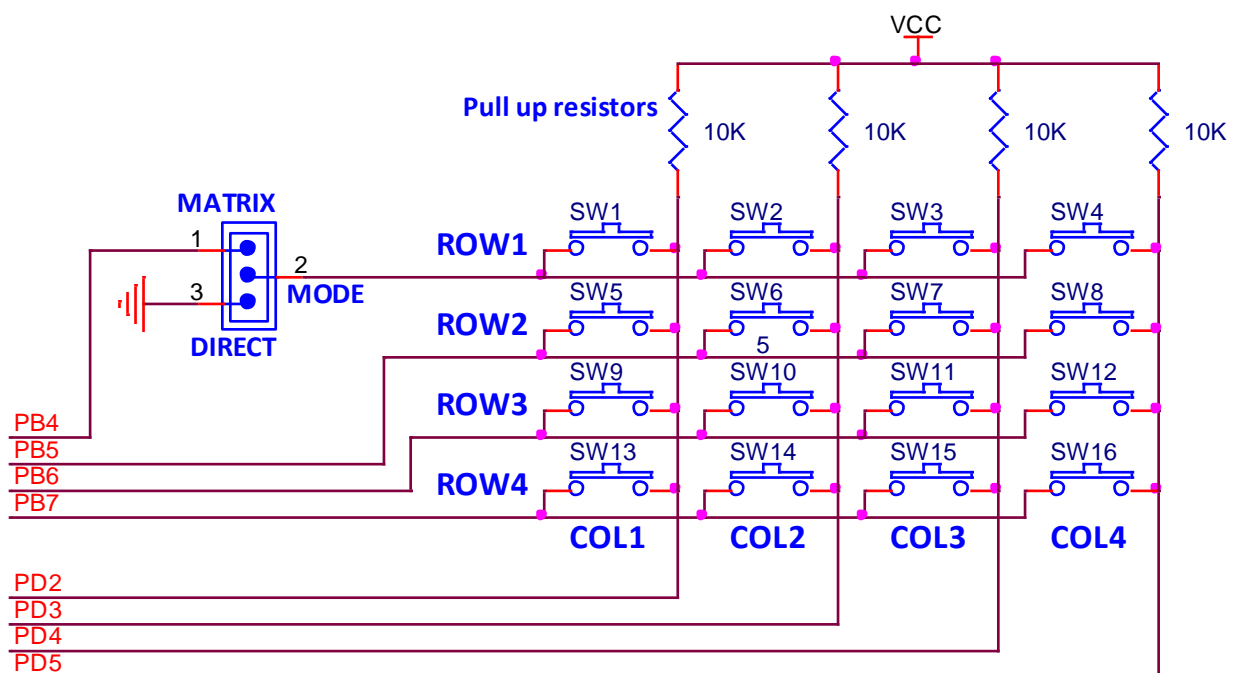
Microcontroller unit (schematic)

4X4 Matrix Keypad Unit

Eta32mini Kit includes 16 pushbutton switches. Switches are configured as 4 rows intersected by 4 columns. Each intersection creates a switch position.



4X4 matrix keypad unit




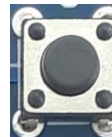














4X4 matrix keypad unit (schematic)

Rows and Columns are connected to microcontroller as following.

| Row number | Microcontroller pin | Arduino pin |
|------------|---------------------|-------------|
| ROW 1 | PB4 | 4 |
| ROW 2 | PB5 | 5 |
| ROW 3 | PB6 | 6 |
| ROW 4 | PB7 | 7 |

| Column number | Microcontroller pin | Arduino pin |
|---------------|---------------------|-------------|
| COL 1 | PD2 | 10 |
| COL 2 | PD3 | 11 |
| COL 3 | PD4 | 12 |
| COL 4 | PD5 | 13 |

| | | | | | |
|---------------|------------|---|---|--|---|
| ROW | ROW1 (PB4) | 1  | 2  | 3  | 4  |
| | ROW2 (PB5) | 5  | 6  | 7  | 8  |
| | ROW3 (PB6) | 9  | 10  | 11  | 12  |
| | ROW4 (PB7) | 13  | 14  | 15  | 16  |
| | | COL1 (PD2) | COL2 (PD3) | COL3 (PD4) | COL4 (PD5) |
| COLUMN | | | | | |

Keypad can be configured as 4X4 matrix keypad (16 switches used) or as 4 direct switches.

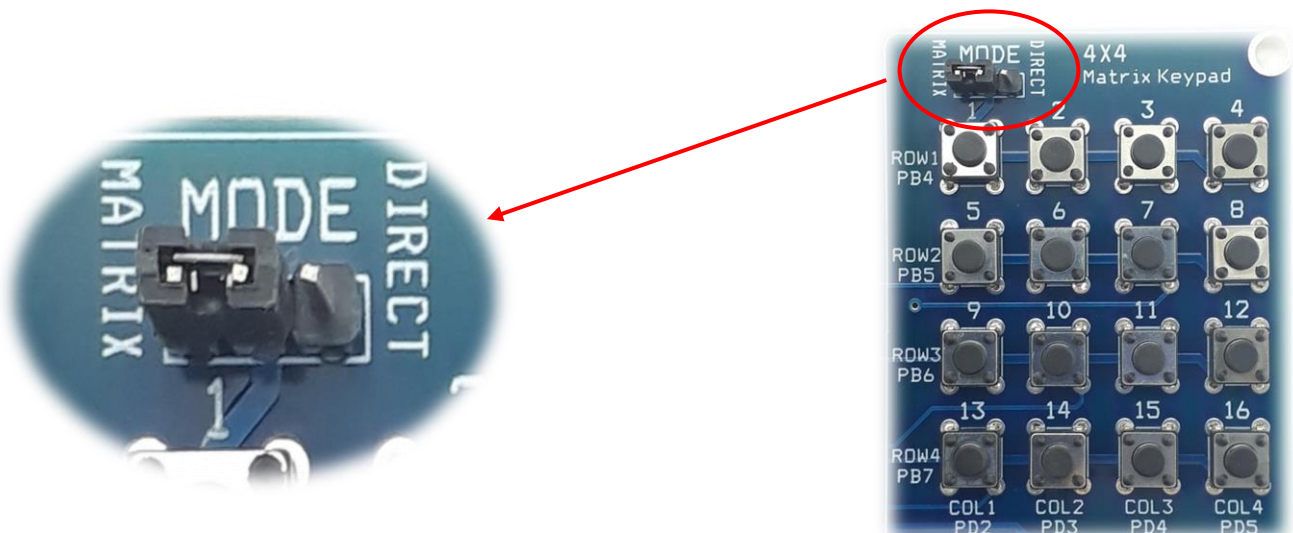
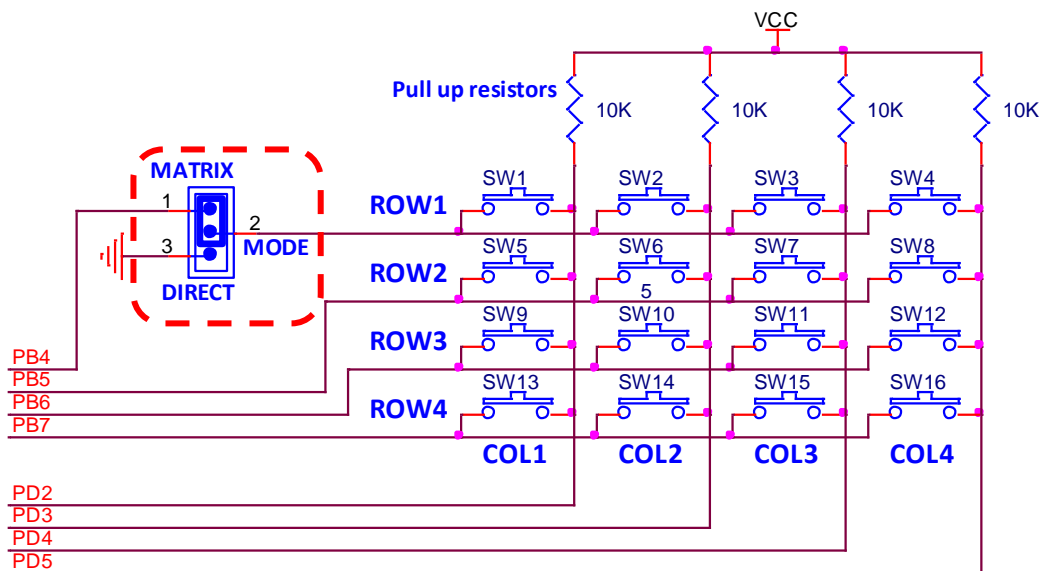
Matrix Mode

In matrix mode, all 16 switches are activated and configured as four row lines and four column lines. Microcontroller scans these lines to detect a button pressed state. Column lines are pulled up by 10K Ω resistors. (i.e. microcontroller port pin reads high if no switch is pressed).

Scan process starts by setting all rows and columns as inputs.

To scan switches in a row, microcontroller configures it as output and sets it to low, then checks columns one at a time. If a column line goes low, microcontroller detects a pressed switch, otherwise, no pressed switch and goes to scan next row and so on.

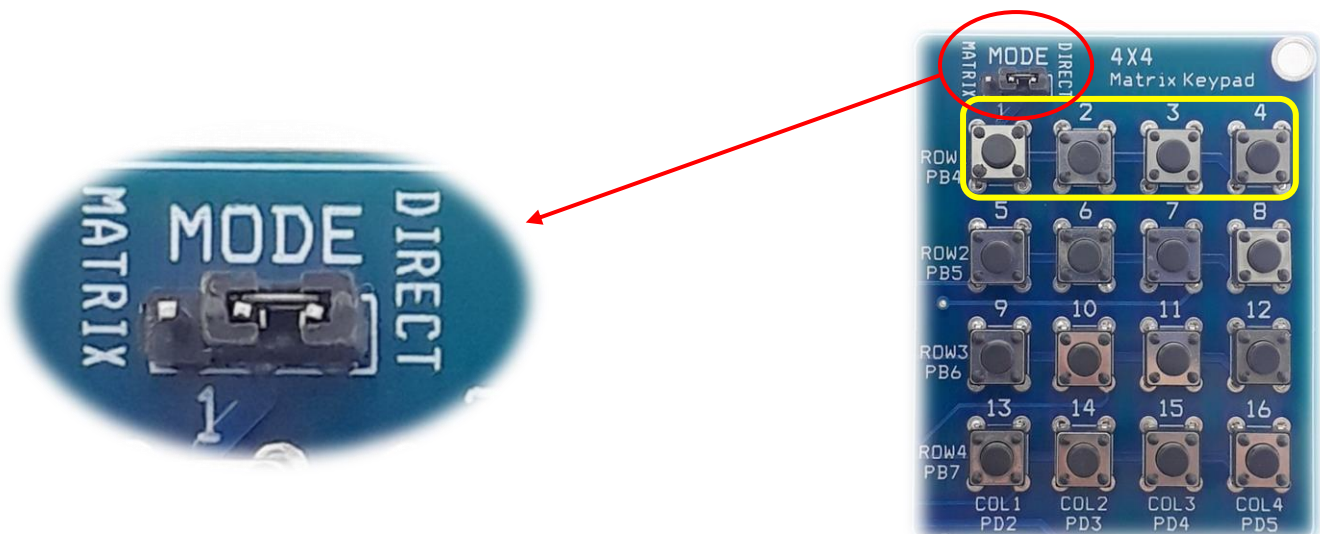
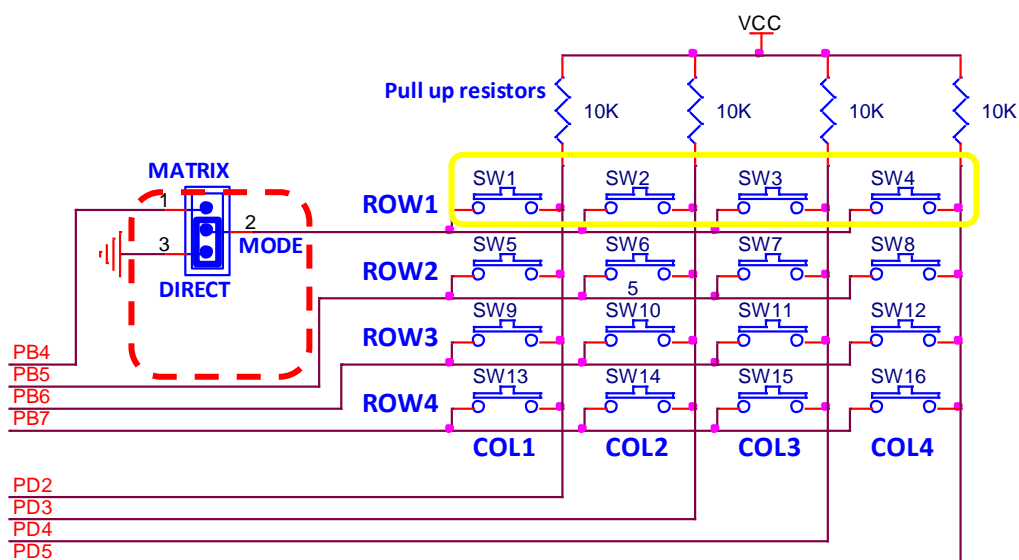
To set keypad in matrix mode, just set MODE jumper to Matrix position as shown below.



Direct Mode

In direct mode, just 4 switches are activated to save microcontroller pins in case of no need for more switches and reducing firmware complexity in scanning operation. Switches in first row only is enabled (i.e. SW1, SW2, SW3 and SW4) and can be read directly as inputs through column lines. Microcontroller checks the column lines. If a column line goes low, it means a switch is pressed.

To set keypad in direct mode, just set MODE jumper to direct position, as shown below:

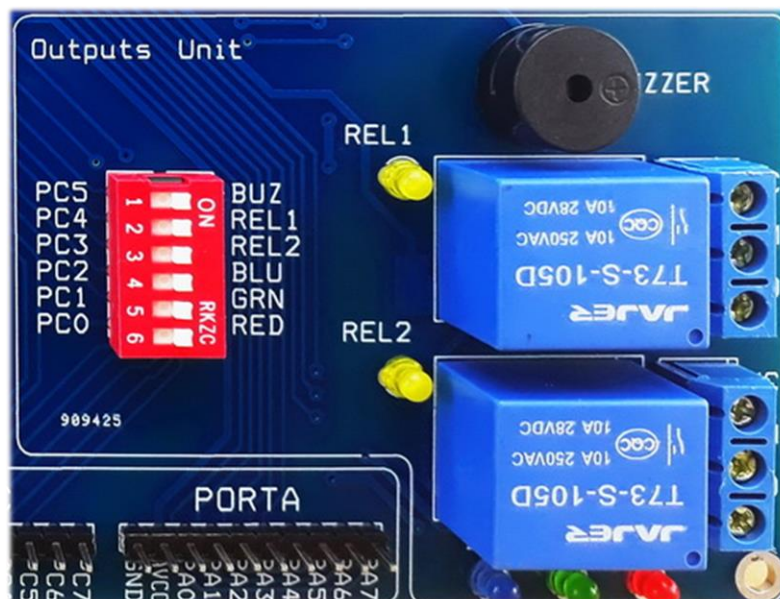


Output Unit

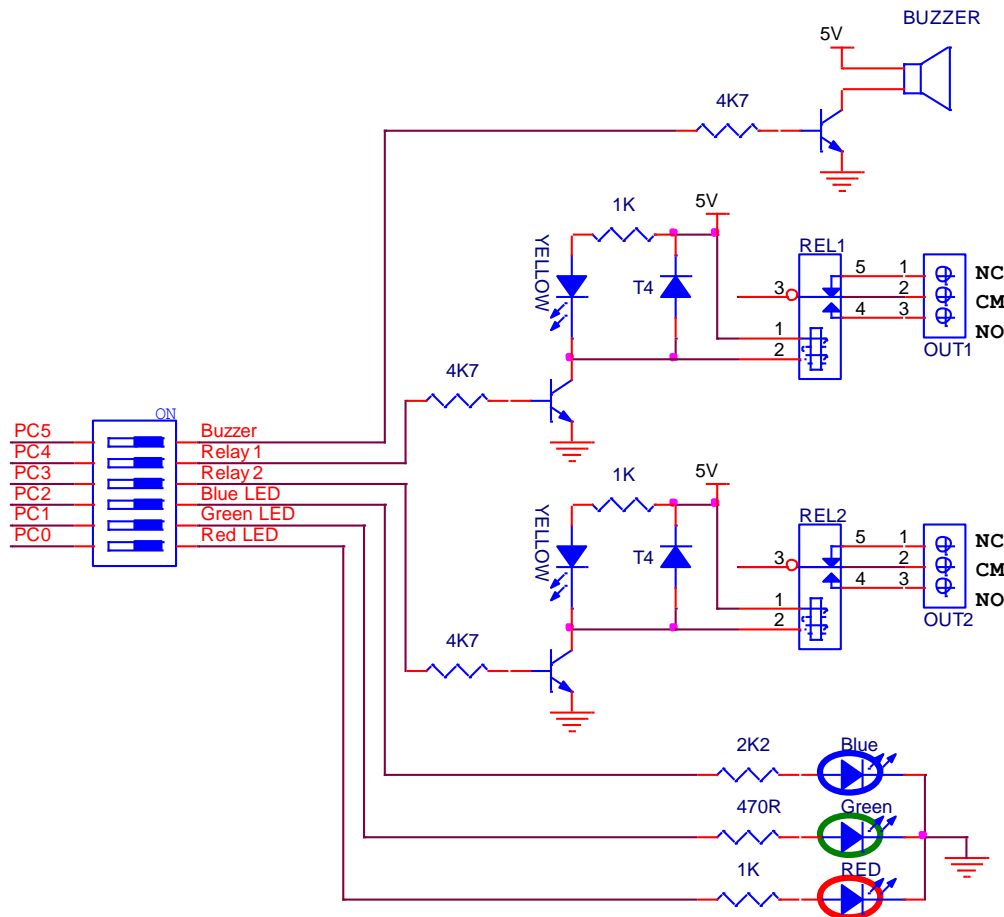
This unit contains six outputs distributed as three LEDs, two relays and one Buzzer.

Outputs are connected to microcontroller as following:

| Output | Microcontroller pin | Arduino pin |
|-----------|---------------------|-------------|
| Buzzer | PC5 | 21 |
| Relay1 | PC4 | 20 |
| Relay2 | PC3 | 19 |
| Blue LED | PC2 | 18 |
| Green LED | PC1 | 17 |
| Red LED | PC0 | 16 |



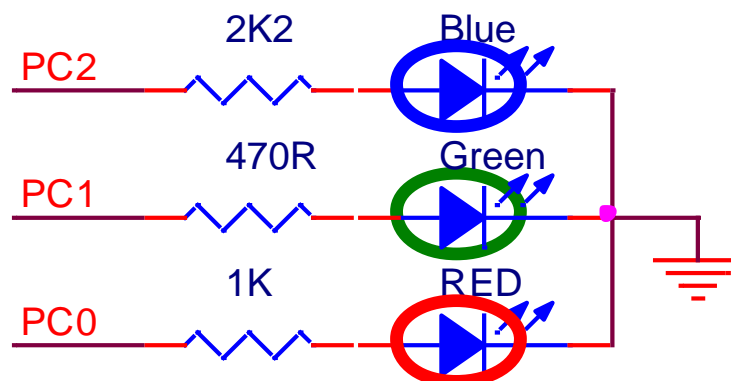
Output unit



Output unit (Schematic)

Three output LEDs

Three LEDs with current limiting resistors are connected to port pins (**PC0**, **PC1** and **PC2**). Red, Green and Blue LED.



Blue LED is connected to **PC.2** in microcontroller.

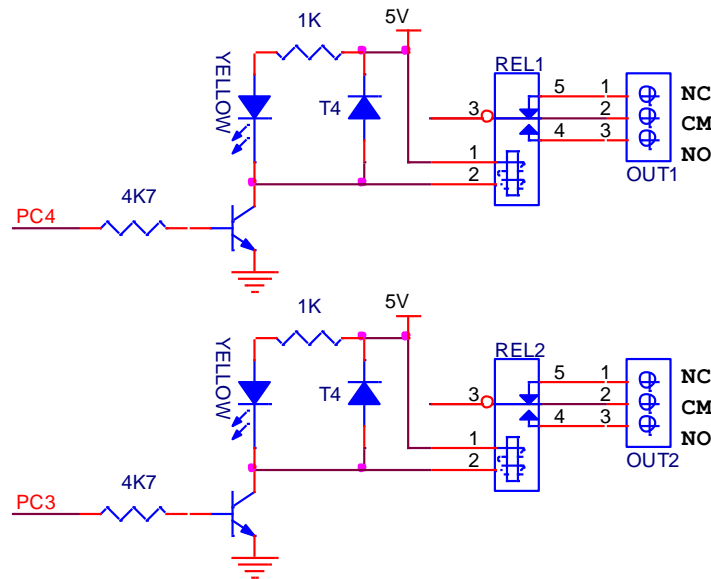
Green LED is connected to **PC.1** in microcontroller.

Red LED is connected to **PC.0** in microcontroller.

Each LED can be enabled or disabled via DIP switch. LEDs are active high. i.e. output high turns LED on and output low turn it off.

Two output relays

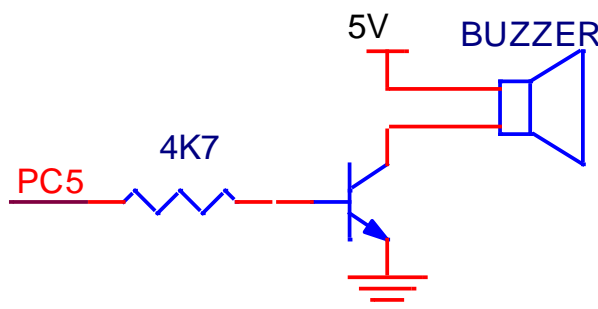
Two output relays are added to *Eta32mini* kit to allow dry contact switches, which is suitable for AC or DC switching applications. Each relay has its own related LED for status indication and can be individually enabled via DIP switch. Relays are driven from NPN transistors. Freewheeling diodes are included to protect transistors from back EMF voltage that arises on relay coil during switching off.



Relay1 is attached to port pin **PC4** and relay2 is attached to port pin **PC3**. Relays are 5V coil and rated to 5A contacts (resistive load). Both of normally open and normally closed contacts are brought out via screw clamp terminals.

One output buzzer

One output buzzer (6VDC) is included in output unit to port pin **PC5**. Also, it may be enabled using DIP switch.



7Segment Display

7segment display is used to indicate numerical data. It can display digits from 0 to 9 and some characters such as A,B,C,H,E,F ,etc.

7segment display is very popular and has many applications.

Eta32mini includes multiplexed two digits 7segment display.

Multiplexed Two Digits 7seg Display Unit

Two multiplexed 7seg digits are added to *Eta32mini* kit. 7segments are referred to by letters "a", "b", "c", "d", "e", "f", "g", "dot". Both digits share the same segments. i.e. segment "a" is the same for DIG1 and DIG2. Each digit has its own common, so there are two commons for the two digits. 7segment display is common cathode, so a high on a segment turns it ON while low turns it OFF. Commons are driven from NPN transistors .So, a high from microcontroller turns transistor on, which enables the digit.

Segments commons are connected to microcontroller as shown in table:

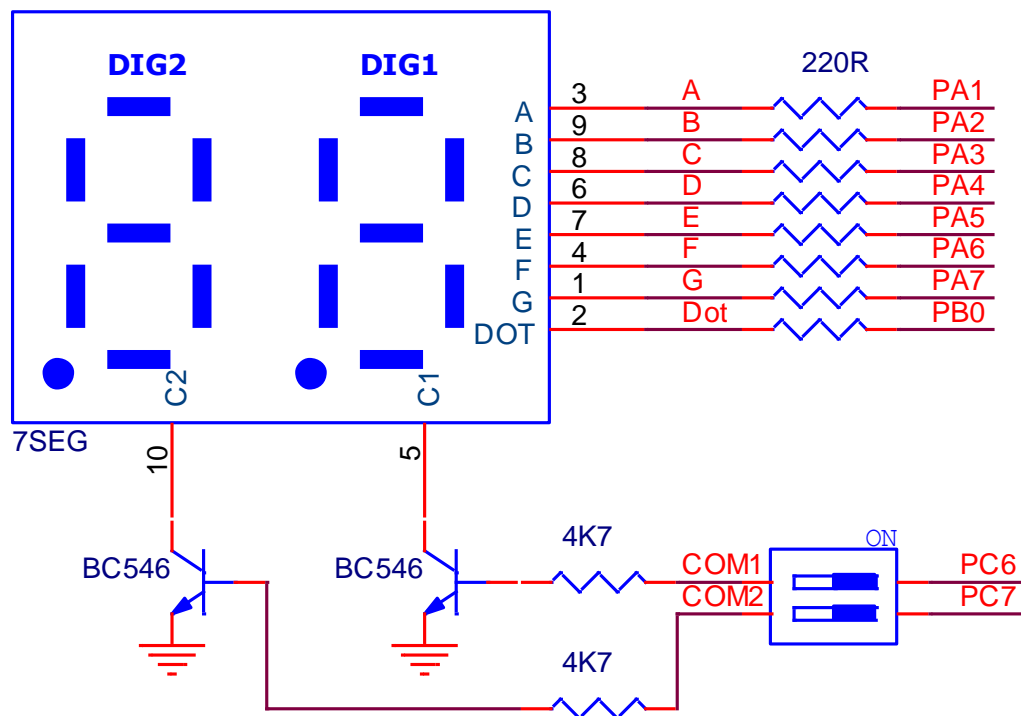
| Segment | microcontroller Pin | Arduino Pin |
|---------|---------------------|-------------|
| A | PA1 | 25 |
| B | PA2 | 26 |
| C | PA3 | 27 |
| D | PA4 | 28 |
| E | PA5 | 29 |
| F | PA6 | 30 |
| G | PA7 | 31 |
| Dot | PB0 | 0 |

7Seg Commons are connected to microcontroller as shown in table:

| Common | microcontroller Pin | Arduino Pin |
|---------|---------------------|-------------|
| Digit 1 | PC6 | 22 |
| Digit 2 | PC7 | 23 |

Note:-

- Each 7seg digit can be enabled or disabled individually using DIP switch.
- 7seg module and 2X16LCD share the same port. So, disable LCD before using 7Segments display.

**7Segment display****7Segment display (schematic)**

2X16 LCD Display Unit

2X16 alphabetic LCD with contrast control configured in 4 bit mode is connected to **PORTA** via DIP switch as following:

LCD module pin out number, symbol and function:

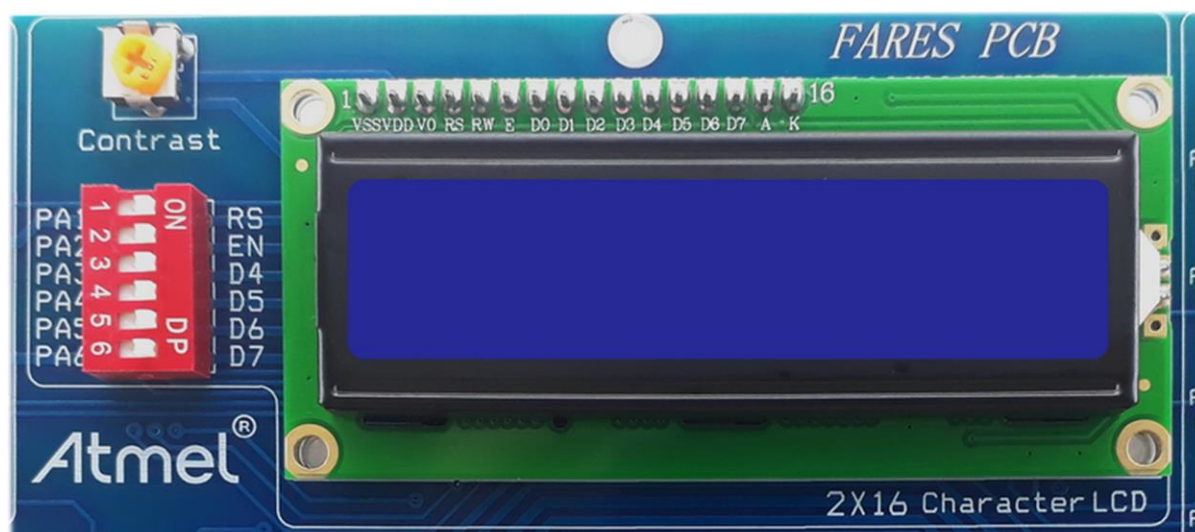
| LCD pin number | LCD pin symbol | LCD pin function |
|----------------|----------------|-------------------------------------|
| 1 | VSS | Ground |
| 2 | VCC | +5V |
| 3 | VO | Contrast adjustment |
| 4 | RS | Register Select(0:Command , 1:Data) |
| 5 | R/W | R/W(0:Write , 1:Read) |
| 6 | EN | Enable |
| 7 | D0 | Data bit 0 |
| 8 | D1 | Data bit 1 |
| 9 | D2 | Data bit 2 |
| 10 | D3 | Data bit3 |
| 11 | D4 | Data bit4 |
| 12 | D5 | Data bit5 |
| 13 | D6 | Data bit6 |
| 14 | D7 | Data bit7 |
| 15 | A | Back light anode(+) |
| 16 | K | Back light cathode(-) |

LCD connection to microcontroller port pins:

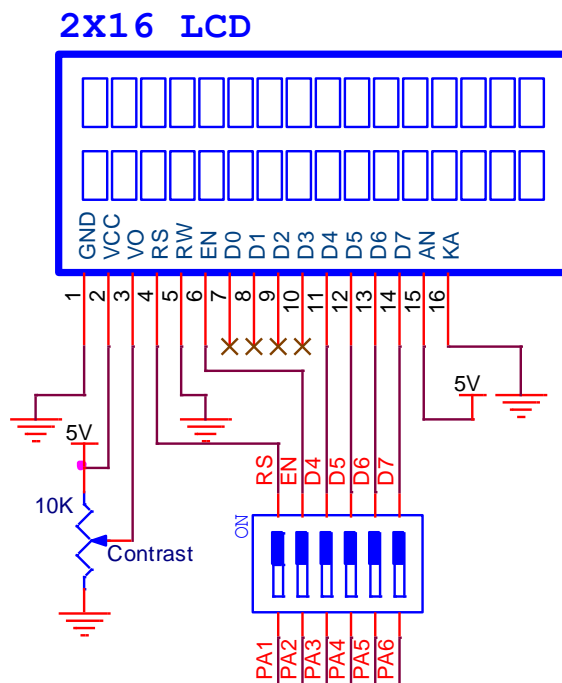
| LCD pin number | LCD pin symbol | Microcontroller pin | Arduino Pin |
|----------------|----------------|---------------------|-------------|
| 4 | RS | PA1 | 25 |
| 6 | EN | PA2 | 26 |
| 11 | D4 | PA3 | 27 |
| 12 | D5 | PA4 | 28 |
| 13 | D6 | PA5 | 29 |
| 14 | D7 | PA6 | 30 |

Note:

- LCD R/W signal is tied to ground.
- 10K Ω variable resistor labeled "Contrast" is adjusted to control the LCD contrast.
- Disable 7Segment module before using LCD module.



2X16 LCD display unit



2X16 LCD display unit (Schematic)

Analog Input Unit

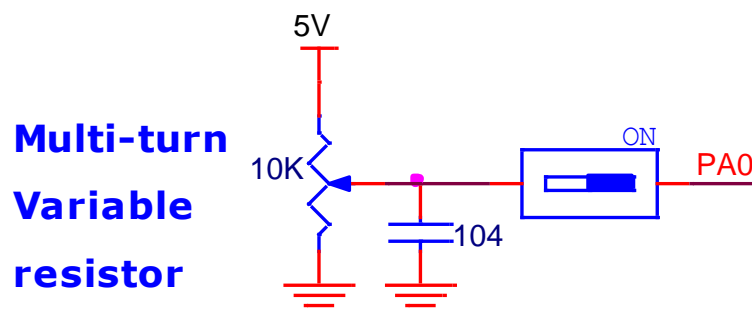
Eta32mini includes 10K Ω multi-turn variable resistor. The fixed terminals of variable resistor are connected to GND and +5VDC whereas the variable terminal is connected via DIP switch to Port pin **PA0** of the microcontroller. The resistor may be adjusted precisely to the required voltage (0.00V to 5.00V).

Analog input connected to microcontroller as shown in table:

| Analog Input | microcontroller Pin | Arduino Pin |
|-------------------|---------------------|-------------|
| Variable resistor | PA0 | 24/A0 |



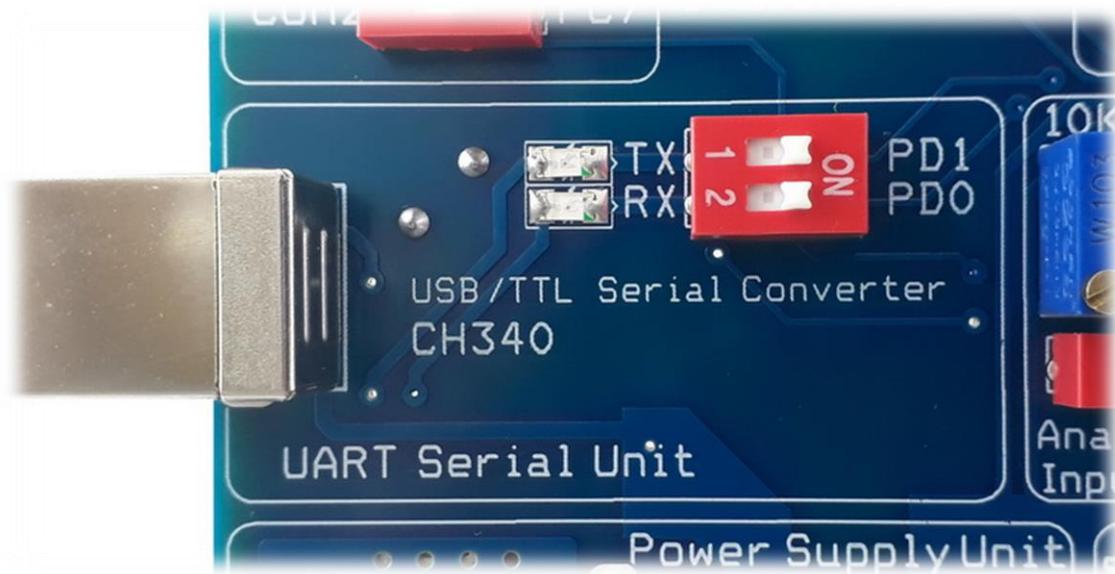
Analog input unit



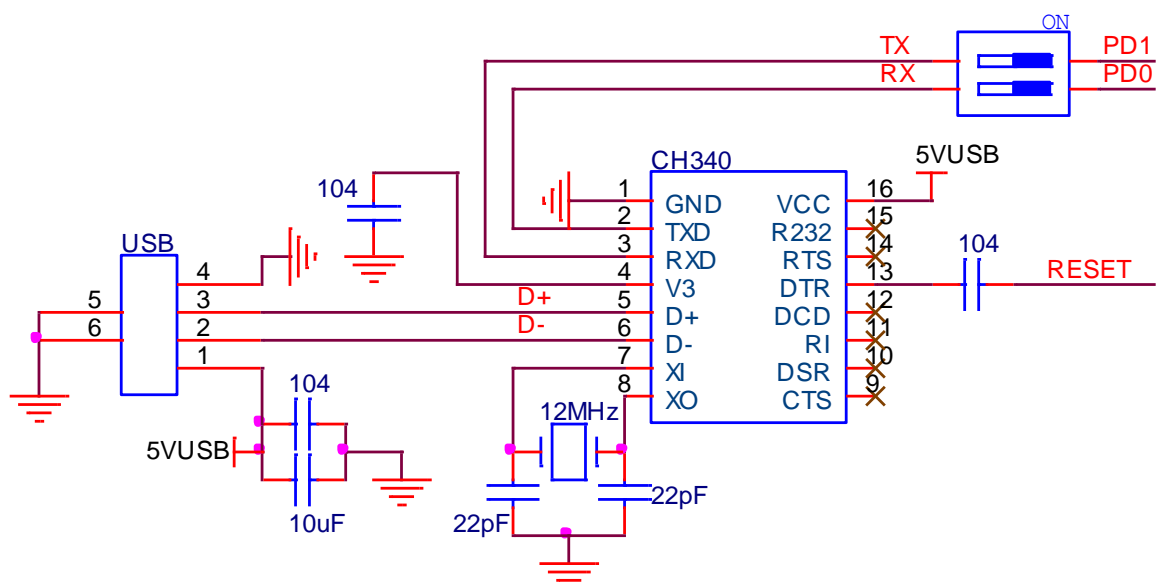
Analog input unit (schematic)

USB Virtual COM Unit

This unit includes USB/TTL logic converter using CH340 IC and provides USB type B socket for PC serial communication. Serial interface circuit can be enabled or disabled using DIP switch (see figure 25). USB/TTL Converter IC is connected to ATMEGA32 microcontroller port pins **PD0 (RX)** and **PD1 (TX)** via DIP switch.



Serial unit



Serial unit (schematic)

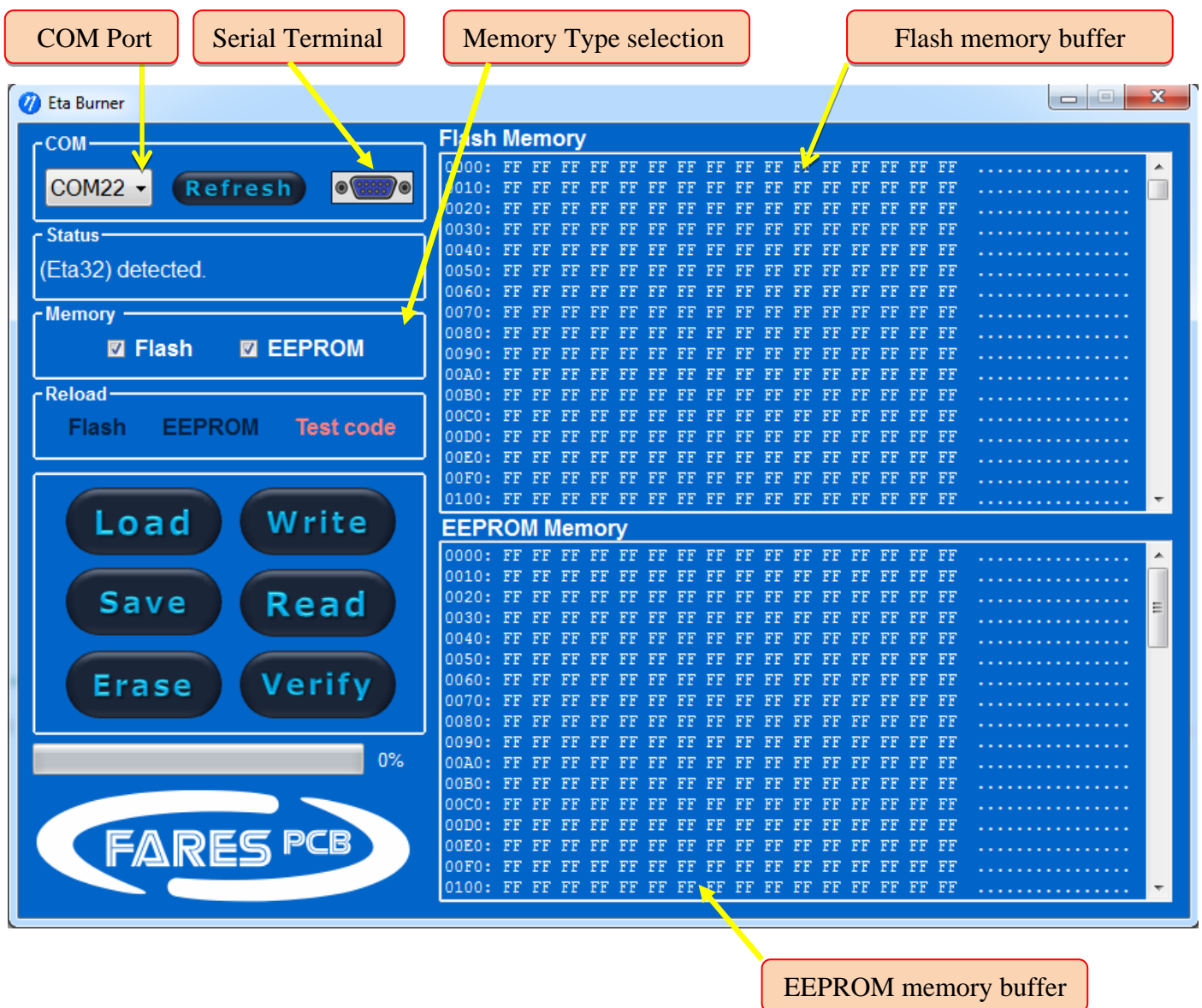
Uploading Hex Code

There are three ways to burn hex code on *Eta32mini* kit.

- 1) *Eta32 Burner* software tool (Eta32mini bootloader from FARESPCB)
- 2) Arduino IDE.
- 3) External programmer.

Programming using *Eta32 Burner*

Eta32mini does not require any sort of external programmers to be programmed. *Eta32mini* kit comes with Eta32mini bootloader, which interfaces an *Eta32 Burner* software tool (from FARESPCB). *Eta32 Burner* software tool is used to download/upload hex code from/to ROM and EEPROM contents of ATMEGA32 microcontroller.



Eta32 Burner tool is developed specifically for *Eta* kits. It detects the kit automatically just after opening it or by clicking refresh button. Select memory type (Flash, EEPROM or both) before any programming operation.

Use "Load" button to load hex file to buffer related to selected memory.

Use "Save" button to store data in the buffer related to selected memory into a hex file (Intel format).

Use "Write" button to burn loaded hex file to selected memory on microcontroller.

Use "Read" button to read hex code from selected memory of microcontroller.

Use "Erase" button to erase selected memory of microcontroller.

Use "Verify" button to compare hex code from selected memory of microcontroller to the related memory buffer.

Fuse settings

By default, fuse bytes are set to the proper values as shown in table

| Fuse byte | Hex value |
|-----------|-----------|
| Low | 9F |
| High | CA |

Brown-out Detector(BOD): Enabled (2.7V).

Clock Source: External crystal oscillator.

Boot reset vector is selected. Bootloader occupies 2K bytes from flash memory

Note:

If fuse bytes are changed to any values other than those shown in the previous table, they must be reprogrammed to the proper values before using *Eta32 Burner* programming tool.

After programming is completed, a reset operation is performed to ensure microcontroller correct operation.

Warning:

Random fuse settings changing is risky. You should take special care while changing these settings. Incorrect fuse settings may cause incorrect microcontroller functioning.

Programming using Arduino IDE

If Arduino software tool is your preferred IDE, follow the next steps to add **Eta32mini** kit to Arduino boards,

1 - Upload Arduino bootloader firmware (ArduinoBL.hex) included in CD using any external programmer such as USBasp programmer. (See programming using external programmer).
Fuse Bytes must be set as shown in table below

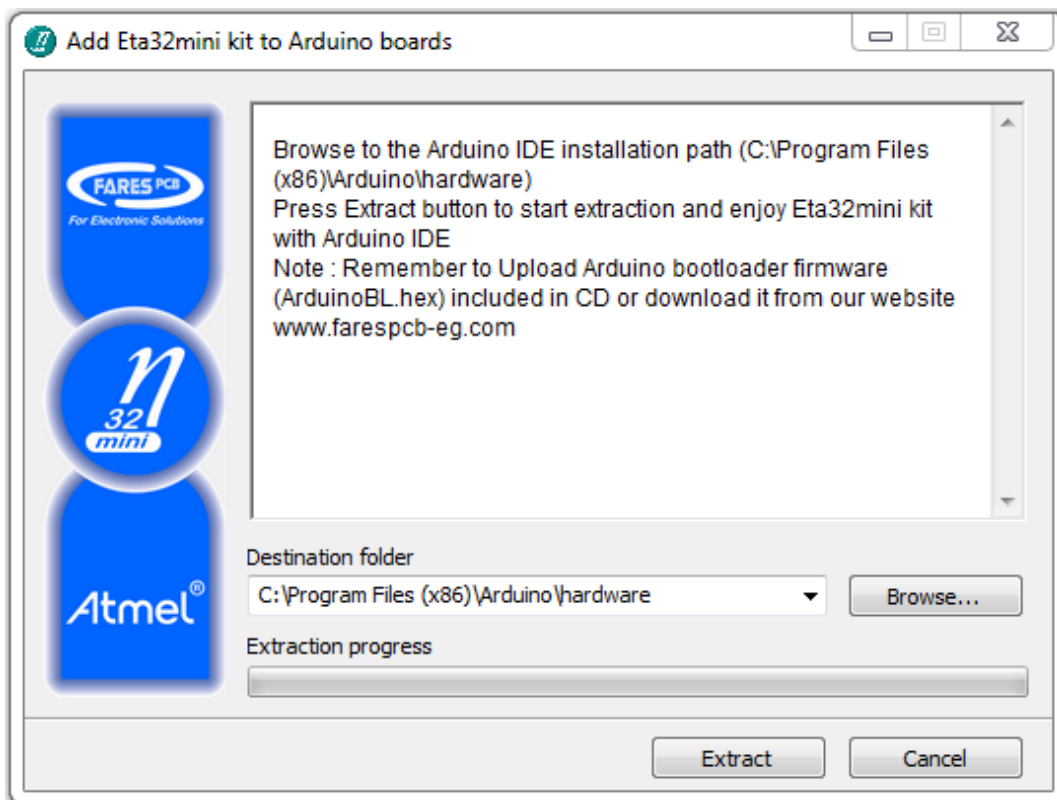
| Fuse byte | Hex value |
|-----------|-----------|
| Low | 9F |
| High | CE |

2 - Download Arduino IDE from the following link

<https://www.arduino.cc/en/Main/Software>

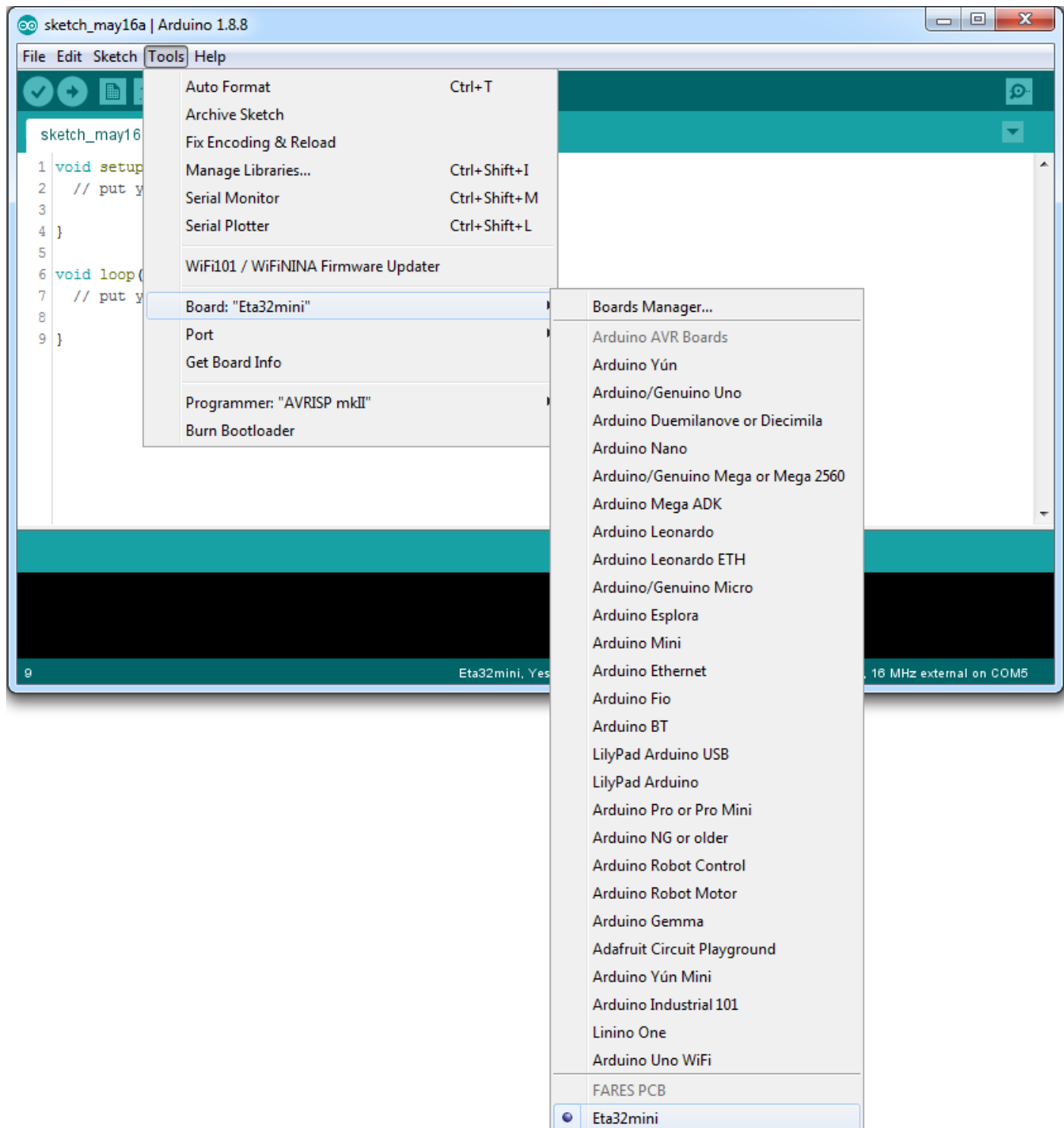
3 - Close Arduino IDE.

4 - Run the self-extracted file(**Eta32miniDuino.exe**)included in CD, and browse to the Arduino IDE installation path (C:\Program Files (x86)\Arduino\hardware), and press Extract button to start extraction



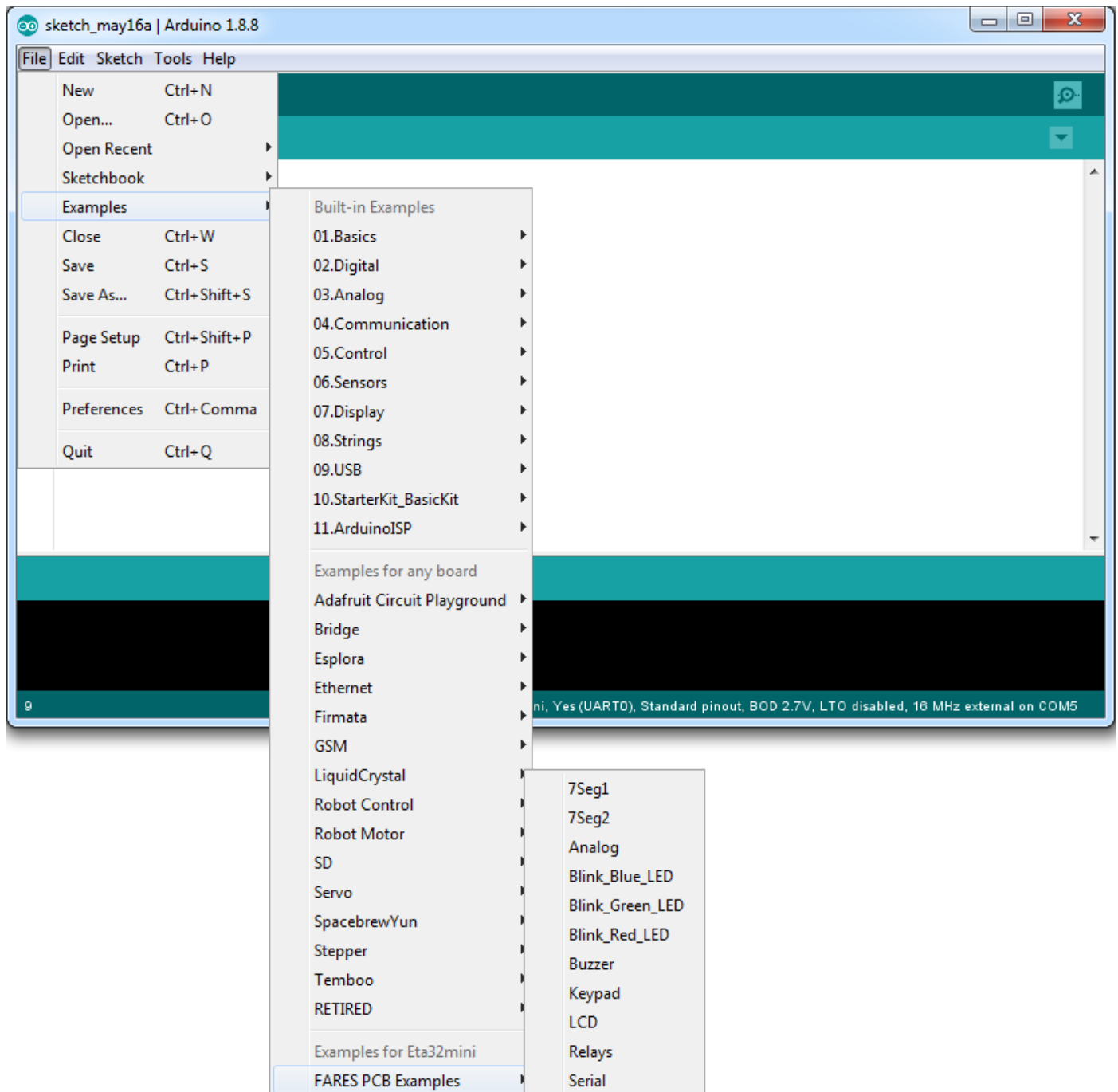
5 –After extraction, *Eta32mini* board is added to Arduino Boards. You can select Eta32mini board as following:

Tools >> Board:>> Eta32mini.



6 - Also, you will find many Arduino example codes to test all units in kit. You can go to:

File >> Examples >> Examples for Eta32mini.



Programming using external programmer

Although *Eta32mini* is designed mainly to be programmed using bootloader. However, it offers ICSP socket to enable programming using any external programmer that supports standard 6 pin ICSP socket, such as USBasp programmer. Use external programmer to download your own application code or even a bootloader firmware such as Arduino bootloader.



← Standard 6 pin ICSP header socket

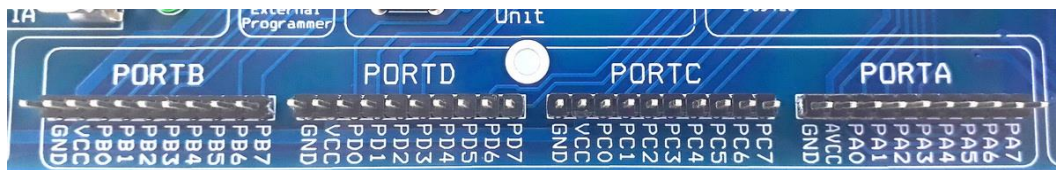
For more details about programming using USBasp programmer, please refer to USBasp CD from FARESPCB products from this link:

<https://fares-pcb.com/product/usbasp-avr-programmer/>

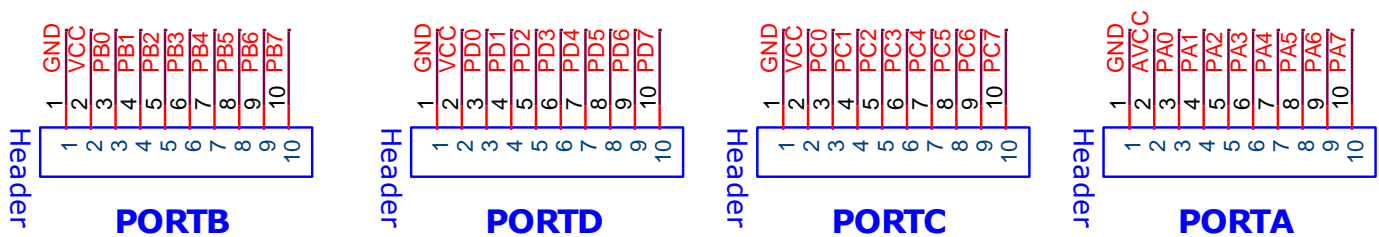
I/O PORT

In addition to the popular built-in circuits and devices included in *Eta32mini* kit, all microcontroller port pins are brought out for external using via header sockets.

External connectors are grouped into 4 units. Each represents one microcontroller port (8 I/O), in addition to GND and 5V.



Microcontroller port pins



Microcontroller port pins (Schematic)

Note:

Please ensure to disconnect the built-in circuits attached to the port pins intended to use.

HOW TO START?

Step1

Install USB driver for ch340.

Step2

Install *Eta32mini* Software tool.

Step3

Connect USB cable to *Eta32mini* kit and open *Eta32mini* software tool. *Eta32mini* tool starts to search COM ports and detect kit automatically. If there's a problem in connection, please review USB cable connection to kit and click "Re-Connect" button to try again.

Step4

Now you are ready to upload your hex code and enjoy working with *Eta32mini* kit. Refer to Programming Using Eta32 Burner for more details about using Eta32 Burner IDE.

If it is the first time to use *Eta32mini* kit, you should perform some test operations on kit before start working on it. The CD included with package contains the firmware code required for testing all modules in kit. So, it's recommended to upload this test code before going to your own application firmware to ensure correct functioning.

Upon burning "**Eta32mini_Test.hex**" code user could test

- Keypad in matrix mode.
 - 2X16 LCD.
 - Outputs unit (Relays, LEDs and Buzzer).
 - 7segments.
 - UART Serial operation.
 - Analog Input.
-
- Enable DIP switches of all modules on kit.
 - Set keypad to matrix mode (Set KB Mode to Matrix position).
 - Plug in USB cable.
 - Turn on power switch.
 - Open *Eta32 Burner* programming tool.
 - Click "LOAD" button to load the test code included in CD (**Eta32mini_Test.hex**).
 - Click Write button.
 - After programming is completed the microcontroller is reset automatically.

After power on or reset operation the test sequence is

1 - Serial module transmits this message to serial port

"FARESPCB Co."

"ATMEL AVR Development kit."

"Eta32minikit."

"Arduino IDE compatible."

User may receive this statement by Hyper Terminal program or any other serial monitor software.

2 - LCD displays the following messages one by one.



FARES PCB Co.
=====



Eta32 mini



ATMEL AVR Kit
Eta32 mini kit

3 - After that this screen is shown



Eta32mini kit
SW -- is Fressed

4 - If a switch is pressed the microcontroller sends a message to serial port contains the pressed switch number and LCD displays the same message such as



Eta32mini kit
SW 01 is Pressed

← When switch "1" is pressed



Eta32mini kit
SW 16 is Pressed

← When switch "16" is pressed

Some switches perform a test operation for specified modules in kit.
These tests are listed in table below

| Switch | Function | |
|--------|---|---|
| | Serial port / LCD (line2) message | Test operation |
| SW1 | "SW 1 is pressed." | ToggleRelay1 status |
| SW2 | "SW 2 is pressed." | Toggle Relay2 status |
| SW3 | "SW 3 is pressed." | Toggle Blue LED |
| SW4 | "SW 4 is pressed." | Toggle Green LED |
| SW5 | "SW 5 is pressed." | Toggle Red LED |
| SW6 | "SW 6 is pressed." | Digits counts from 00 to 99 and perform auto reset |
| SW7 | "SW 7 is pressed." | LCD displays the analog volt measured on port pin (PA0) |
| SW8 | "SW 8 is pressed." | LCD displays any received text from the serial port in the second line. |
| SW9 | "SW 9 is pressed." | --- |
| SW10 | "SW10 is pressed." | --- |
| SW11 | "SW11 is pressed." | --- |
| SW12 | "SW12 is pressed." | --- |
| SW13 | "SW13 is pressed." | --- |
| SW14 | "SW14 is pressed." | --- |
| SW15 | "SW15 is pressed." | --- |
| SW16 | "SW16 is pressed." | --- |

Special Thanks to:

Eng. / Ahmed Ibrahim Ahmed

ah.ahmed@nu.edu.eg**Copyright © 2019 by FARESPCB**For our full range of products see our website at <http://www.fares-pcb.com>

If you have any technical questions about our products, e-mail us at

www.support@fares-pcb.com**FARESPCB co. (Head office)**

164 Tahrir st,
Bab El-Louq,
Cairo,
Egypt.
Tel: +202-23904484
Mob:+201000652977

FARESPCB co. (Branch office)

4 El-Shabrawy st,
Road El-Farag,
Cairo,
Egypt.
Tel: +202-24577118
Mob:+201022457902

FARESPCB Co reserves the right to make changes in circuit design, software and/or specifications at any time without prior notification. For the most up-to-date information, please visit our web site at <http://www.fares-pcb.com>

Information furnished by FARESPCB is believed to be accurate and reliable. However, FARESPCB assumes no responsibility arising from the use of the specifications described.

Warrantee: FARESPCB™ warrants its products against defects in materials and workmanship for a period of 30 days. If you discover a defect, we will, at our option, repair or replace your product or refund your purchase price. This warrantee does not cover products that have been physically abused or misused in any way.

Distributor:

RAM Electronics
32 El Falaky St. Bab El Louk
Tahrir, Cairo
Egypt.
Tel: +202-27960551
www.ram.com.eg
Sales@ram-electronics.com

