ELECTRONIC VOTING MACHINE WITH FINGERPRINT AND FACIAL RECOGNITION

A PROJECT REPORT

Submitted by

AJAY. R	**(620118106003)**
BHARATH.G	**(620118106015)**
KALAIARASAN.M. K	**(620118106034)**
MUBARAK ALI.S. Y	**(620118106056)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONIC AND COMMUBICATION ENGINEERING

AVS ENGINEERING COLLEGE, SALEM

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "…**ELECTRONIC VOTING MACHINE WITH FINGERPRINT AND FACIAL RECOGNITION…"** is the bonafide work of "…**AJAY.R (620118106003), BHARATH.G (620118106015), KALAIARASAN M.K (620118106034), MUBARAK ALI S.Y (620118106056) …**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Prof. RAMYA R S, M.E.,	**Prof. KANAGARAJ G, M.E, Ph.D.,**
ASSOCIATE PROFESSOR	ASSISTANT PROFESSOR
Department of ECE, AVS Engineering College, Salem-636003.	Department of ECE, AVS Engineering College, Salem-636003.

Submitted for Project Viva -Voice Examination held on_______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

A great deal of time and effort has been spent in completing this project work. Several people have guided us and contributed significantly to this effort and so this become obligate to record our thanks to them.

We express our profound gratitude to our honorable Chairman, **Shri. K.KAILASAM,** Secretary, **Mr. K.RAJAVINAYAKAM, MBA.,** and also to our Correspondent, **Mr. K.SENTHILKUMAR, B.Tech.,** for providing all necessary facilities for the successful completion of the project.

We have immense pleasure in expressing our gratitude to our beloved Principal, Dr. **A.SRINIVASAN, M.E, Ph.D.,** for allowing us to have extensive use of our college facilities to do this project.

We have immense pleasure in expressing our gratefulness to our beloved Vice Principal, **Dr. R.VISWANATHAN, M.E, Ph.D.,** for allowing us to have extensive use of our college facilities to do this project.

We express our heartiest gratitude to Head of the Department **Prof. R S. RAMYA, M.E.,** for her guidance and encouragement.

We express our sincere and heartiest gratitude to class advisor, **Prof. G. KANAGARAJ M.E, Ph.D.,** for his constant support and excellent guidance in all aspects.

We indebted to our project coordinator and our guide, **Prof. G. KANAGARAJ, M.E, Ph.D.,** for her constant help and excellent creative ideas over the period of project work.

I specially thank for all our Teaching and Non-Teaching Staff Members and Lab Technicians of the Department of Electronics and Communication Engineering for their encouragement to do the project work with full interest and enthusiasm. We pay our profound gratitude to the Almighty God for his invisible vigilance and would like to thank our Parents forgiving us support and encouragement.

DECLARATION

We affirm that the Project Phase-I entitled "**ELECTRONIC VOTING MACHINE WITH FINGERPRINT AND FACIAL RECOGNITION**" being submitted in partial fulfillment for the award of BACHELOR OF ENGINEERING is the original work carried out by me. It has not formed the part of any other Project Phase-I submitted for the award of any degree or diploma, either in this or any other university.

1)

2)

3)

4)

Signature of the Candidate,

AJAI R	**(620118106003)**
BHARATH G	**(620118106015)**
KALAIARASAN M K	**(620118106034)**
MUBARAK ALI S Y	**(620118106056)**

I certify that the declaration made above by the candidate is true.

Signature of the Guide,

KANAGARAJ G, M.E, Ph.D.,

Assistant Professor,

Department of ECE,

AVS Engineering College,

Salem-636003.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	1
1	**INTRODUCTION**	2
2	**SYSTEM ANALYSIS**	2
	2.1 EXISTING SYSTEM	2
	2.2 PROPOSED SYSTEM	3
3	**BLOCK DIAGRAM**	3
	3.1 HARDWARE	4
	3.2 SOFTWARE	4
4	**MODULE DESCRIPTOPN**	4
	4.1 POWER SUPPLY	4
	4.1.1 TRANSFORMER	4
	4.1.2 RECTIFIER	5
	4.1.3 FILTER	5
	4.1.4 VOLTAGE REGULATOR	5
	4.2 MICROCONTROLLER – ARDDUINO	4
	4.3 PIN CONFIGURATION	7
	4.4 PIN DESCRIPTION	7
	4.5 FINGERPRINT RECOGNITION SYSTEM	16
	4.5.1 FINGERPRINTS AS USED FOR IDENTIFICATION	16
	4.5.2 FINGERPRINT TYPES	17

5	**SOFTWARE**	19
	5.1 ARDUINO IDE	19
	5.2 ARDUINO IDEDUDE ARDUINO IDE PROGRAMMER	39
	5.2.1 FEATURES	40
	5.2.2 CONNECTION TO THE TARGET MICROCONTROLLER	40
	5.2.3 CONNECTORS	41
6	**CONCLUSION**	42
7	**REFERENCES**	42

ELECTRONIC VOTING MACHINE WITH FINGERPRINT AND FACIAL RECOGNITION

ABSTRACT

The objective of voting is to allow voters to exercise their right to express their choices regarding specific issues, pieces of legislation, citizen initiatives, constitutional amendments, recalls and/or to choose their government and political representatives. It has always been an onerous task for the election commission to conduct free and fair polls in our country, the largest democracy in the world. A lot of money has been spent on this to make sure that the elections are rampage free. But, now- a -days it has become very usual for some forces to indulge in rigging which may eventually lead to a result contrary to the actual verdict given by the people. In order to provide inexpensive solutions to the above, this project will be implemented with biometric system i.e., finger print scanning. This is used to ensure the security to avoid fake, repeated voting etc. It also enhances the accuracy and speed of the process. The system uses thumb impression for voter identification as we know that the thumb impression of every human being has a unique pattern. Thus, it would have an edge over the present-day voting systems. The purpose of such system is to ensure that the voting rights are accessed only by a legitimate user and no one else. In this, creation of a database consisting of the thumb impressions of all the eligible voters in a constituency is done as a pre-poll procedure. During elections, the thumb impression of a voter is entered as input to the system. This is then compared with the available records in the database. If the particular pattern matches with anyone in the available record, access to cast a vote is granted. But in case the pattern doesn't match with the records of the database or in case of repetition, access to cast a vote is denied or the vote gets rejected. The result is instantaneous and counting is done. The overall cost for conducting elections gets reduced and so does the maintenance cost of the systems.

CHAPTER 1

INTRODUCTION

INTRODUCTION

Biometrics is the science and technology of measuring and analyzing biological data. In information technology, biometrics refers to technologies that measure and analyze human body characteristics, such as DNA, FINGERPRINT AND FACE RECOGNITION s, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes. In this paper we have used thumb impression for the purpose of voter identification or authentication. As the thumb impression of every individual is unique, it helps in maximizing the accuracy. A database is created containing the thumb impressions of all the voters in the constituency. Illegal votes and repetition of votes is checked for in this system. Hence if this system is employed the elections would be fair and free from rigging. Thanks to this system that conducting elections would no longer be a tedious and expensive job.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Electronic Voting Machines ("EVM"), Idea mooted by the Chief Election Commissioner in 1977. The EVMs were devised and designed by Election Commission of India in collaboration with Bharat Electronics Limited (BEL), Bangalore and Electronics Corporation of India Limited (ECIL), Hyderabad. The EVMs are now manufactured by the above two undertakings. An EVM consists of two units, I) Control Unit, ii) Balloting Unit. The two units are joined by a five-meter cable. The Control Unit is with the Presiding Officer or a Polling Officer and the Balloting Unit is placed inside the voting compartment.

There are two types of problems with EVM which is currently in use:

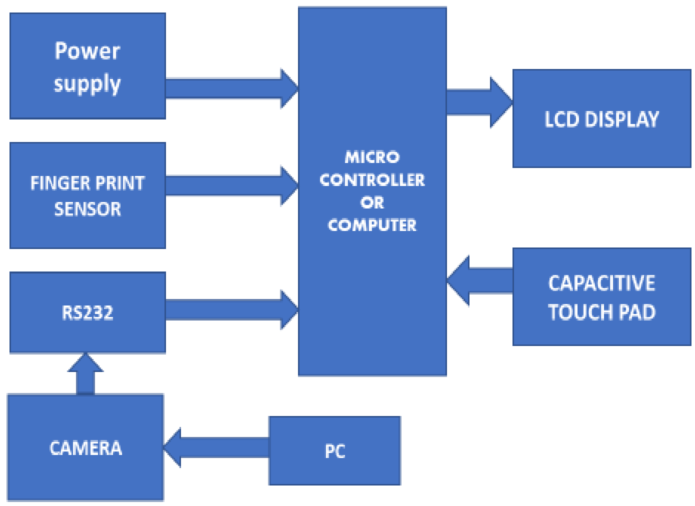
1. Security Problems - One can change the program installed in the EVM and tamper the results after the polling. By replacing a small part of the machine with a look-alike component that can be silently instructed to steal a percentage of the votes in favor of a chosen candidate. These instructions can be sent wirelessly from a mobile phone.
2. Illegal Voting (Rigging) - The very commonly known problem, rigging which is faced in every electoral procedure. One candidate casts the votes of all the members or few amounts of members in the electoral list illegally. This results in the loss of votes for the other candidates participating and also increases the number votes to the candidate who performs this action. This can be done externally at the time of voting.

2.2 PROPOSED SYSTEM

A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database. Depending on the application context, a biometric system may operate either in verification mode or identification mode. In addition, different from the manual approach for FINGERPRINT AND FACERECOGNITION recognition by experts, the FINGERPRINT AND FACE RECOGNITION here is referred as AFRS (Automatic FINGERPRINT AND FACE RECOGNITION System), which is program-based.

CHAPTER 3

BLOCK DIAGRAM



3.1 HARDWARE

- Power supply unit
- FINGERPRINT sensor
- ARDUINO UNO
- TOUCH SENOSR
- BATTERY
- LCD DISPLAY

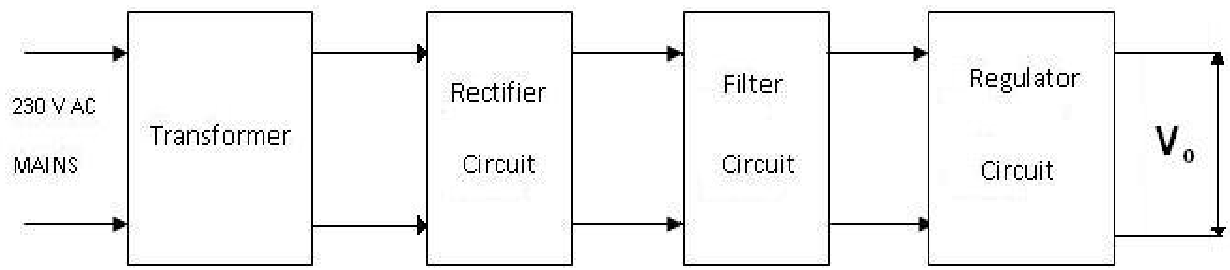
3.2 SOFTWARE

- Mat lab 8.1
- Platform - ARDUINO IDE STUDIO
- In System Programmer – Prog ISP 172
- Compiler – Win ARDUINO IDE

CHAPTER 4

MODULE DESCRIPTION

4.1 Power Supply



The given block diagram includes following:

Transformer: A transformer is an electromagnetic static device, which transfers electrical energy from one circuit to another, either at the same voltage or at different voltage but at the same frequency**.**

Rectifier: The function of the rectifier is to convert AC to DC current or voltage. Usually in the rectifier circuit full wave bridge rectifier is used.

Filters: The Filter is used to remove the pulsated AC. A filter circuit uses capacitor and inductor. The function of the capacitor is to block the DC voltage and bypass the AC voltage. The function of the inductor is to block the AC voltage and bypass the DC voltage.

Voltage Regulator: Voltage regulator constitutes an indispensable part of the power supply section of any electronic systems. The main advantage of the regulator ICs is that it regulates or maintains the output constant, in spite of the variation in the input supply.

4.2 MICROCONTROLLER – ARDUINO

• High-performance, Low-power ARDUINO IDE 8-bit Microcontroller

• Advanced RISC Architecture

- ❖ 130 Powerful Instructions – Most Single-clock Cycle Execution

- ❖ 32 x 8 General Purpose Working Registers
- ❖ Fully Static Operation
- ❖ Up to 16 MIPS Throughput at 16 MHz
- ❖ On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory segments

- ❖ 8K Bytes of In-System Self-programmable Flash program memory
- ❖ 512 Bytes EEPROM
- ❖ 1K Byte Internal SRAM
- ❖ Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- ❖ Data retention: 20 years at 85°C/100 years at 25°C
- ❖ Optional Boot Code Section with Independent Lock Bits
- ❖ In-System Programming by On-chip Boot Program
- ❖ True Read-While-Write Operation
- ❖ Programming Lock for Software Security

• Peripheral Features

- ❖ Two 8-bit Timer/Counters with Separate pre scalar, one Compare Mode
- ❖ One 16-bit Timer/Counter with Separate Pre scalar, Compare Mode, and Capture Mode.
- ❖ Real Time Counter with Separate Oscillator
- ❖ Three PWM Channels
- ❖ 8-channel ADC in TQFP and QFN/MLF package
- ❖ 6-channel ADC in PDIP package
- ❖ Six Channels 10-bit Accuracy
- ❖ Byte-oriented Two-wire Serial Interface
- ❖ Programmable Serial USART
- ❖ Master/Slave SPI Serial Interface
- ❖ Programmable Watchdog Timer with Separate On-chip Oscillator
- ❖ On-chip Analog Comparator

• Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby.

• I/O and Packages

- 23 Programmable I/O Lines
- 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF

• Operating Voltages

- 2.7 - 5.5V (ARDUINOL)
- 4.5 - 5.5V (ARDUINO)

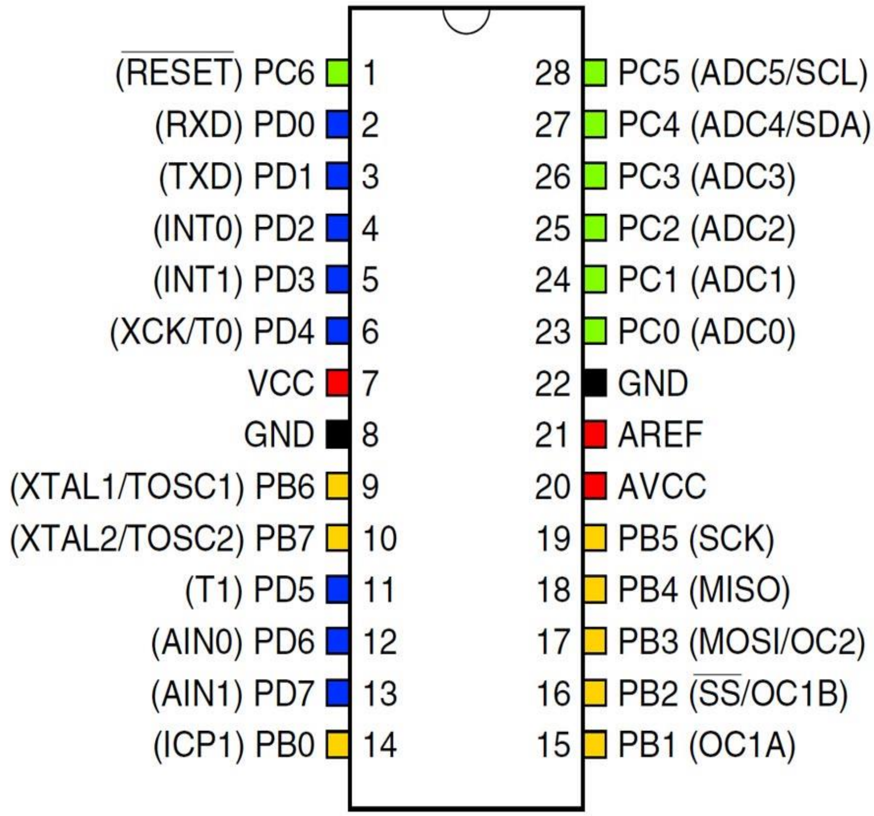
• Speed Grades

- 0 - 8 MHz (ARDUINOL)
- 0 - 16 MHz (ARDUINO)

• Power Consumption at 4 MHz's 3V, 25°C

- Active: 3.6 mA
- Idle Mode: 1.0 mA

4.3 Pin Configurations



4.4 Pin Descriptions

- VCC - Digital supply voltage.
- GND - Ground.
- Port B (PB7-PB0)
- XTAL1/XTAL2/TOSC1/TOSC2

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. Depending on the clock selection, fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7-6 is used as TOSC2-1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC5.PC0)

Port C is a 7-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.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is un programmed, PC6 is used as a 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.

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.

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.

AVCC

AVCC is the supply voltage pin for the A/D Converter, Port C (3-0), and ADC (7-6). 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. Note that Port C (5.4) use digital supply voltage, VCC.

AREF

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

LCD

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.

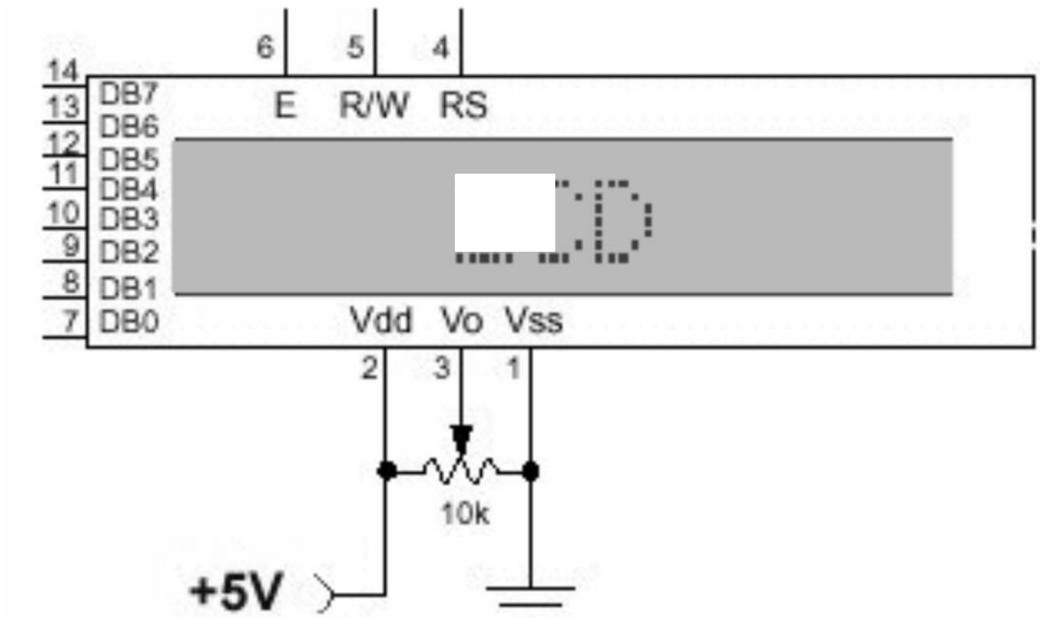


Fig 4.6 Pin Diagram of LCD

Pin configuration

PIN CONNECTIONS			
PIN	**Symbol**	**Level**	**Function**
1	VSS	—	GND(0V)
2	VDD	—	Supply Voltage for Logic(+5V)
3	V0	—	Power supply for LCD
4	RS	H/L	H: Data; L: Instruction Code
5	R/W	H/L	H: Read; L: Write
6	E	H/L	Enable Signal
7	DB0	H/L	Data Bus Line
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	BL1	—	Backlight Power(+5V)
16	BL2	—	Backlight Power(0V)

MAX 232

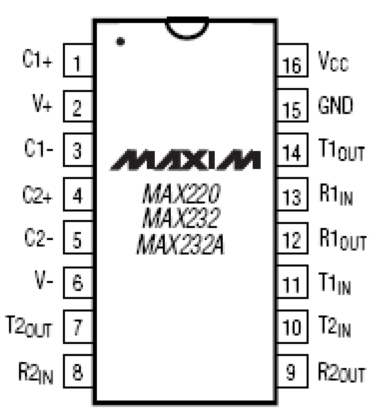


Fig4.3: Pin diagram of MAX232

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available. These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5μW.

RS 232

Due to its relative simplicity and low hardware overhead (as compared to parallel interfacing), serial communications are used extensively within the electronics industry. Today, the most popular serial communications standard in use is certainly the EIA/TIA–232–E specification. This standard, which has been developed by the Electronic Industry Association and the Telecommunications Industry Association (EIA/TIA), is more popularly referred to simply as “RS–232” where “RS” stands for “recommended standard”. In recent years, this suffix has been replaced with “EIA/TIA” to help identify the source of the standard. We use the common notation “RS–232”.

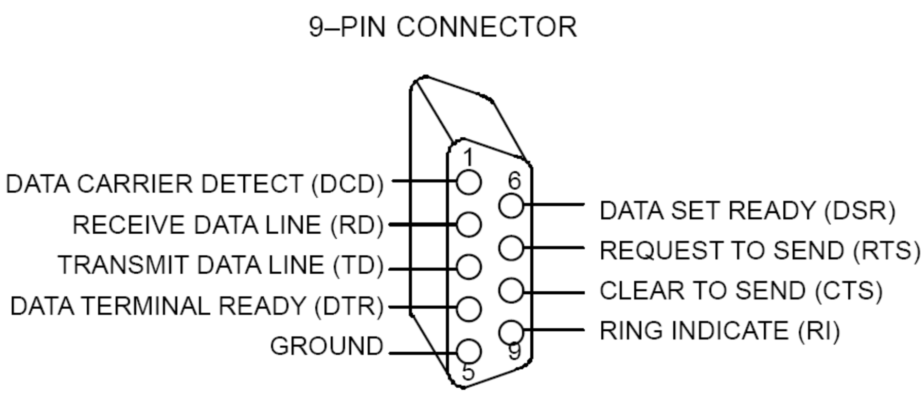


Fig 4.4 DB-9 Connector

4 Liquid-crystal display

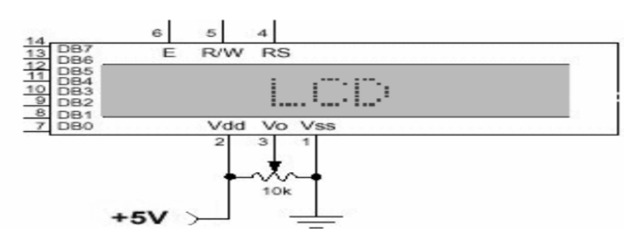


Fig 4.5 Pin Diagram of LCD

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.· LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except those arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

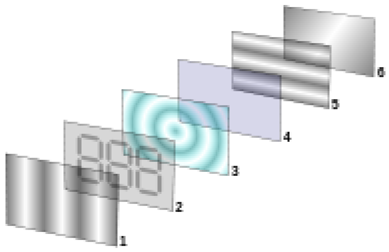
LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big-screen television set.

Since LCD screens do not use phosphors, they do not suffer image burn-in when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft schedule on an indoor sign). LCDs are, however, susceptible to image persistence.· The LCD screen is more energy-efficient and can be disposed of more safely than a CRT can.

Its low electrical power consumption enables it to be used in battery-powered electronic equipment more efficiently than CRTs can be. By 2008, annual sales of televisions with LCD screens exceeded sales of CRT units worldwide, and the CRT became obsolete for most purposes.

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters (parallel and perpendicular), the axes of transmission of which are (in most of the cases) perpendicular to each other. Without the liquid crystal between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer. Before an electric field is applied, the orientation of the liquid-crystal molecules is determined by the alignment at the surfaces of electrodes. In a twisted nematic (TN) device, the surface alignment directions at the two electrodes are perpendicular to each other, and so the molecules arrange themselves in a helical structure, or twist. This induces the rotation of the polarization of the incident light, and the device appears gray. If the applied voltage is large enough, the liquid crystal molecules in the center of the layer are almost completely untwisted and the polarization of the incident light is not rotated as it passes through the liquid crystal layer. This light will then be mainly polarized perpendicular to the second filter, and thus be blocked and the pixel will appear black. By controlling the voltage applied across the liquid crystal layer in each pixel, light can be allowed to pass through in varying amounts thus constituting different levels of gray. Color LCD systems use the same technique, with color filters used to generate red, green, and blue pixels The optical effect of a TN device in the voltage-on state is far less dependent on variations in the device thickness than that in the voltage-off state. Because of this, TN displays with low information content and no backlighting are usually operated between crossed polarizers such that they appear bright with no voltage (the eye is much more sensitive to variations in the dark state than the bright state). As most of 2010-era LCDs are used in television sets, monitors and smartphones, they have high-resolution matrix arrays of pixels to display arbitrary images using backlighting with a dark background. When no image is displayed, different arrangements are used. For this purpose, TN LCDs is operated between parallel polarizers, whereas IPS LCDs feature crossed polarizers. In many applications IPS LCDs have replaced TN LCDs, in particular in smartphones such as iPhones. Both the liquid crystal material and the alignment layer material contain ionic compounds. If an electric field of one particular polarity is applied for a long period of time, this ionic material is attracted to the surfaces and degrades the device performance. This is avoided either by applying an alternating current or by reversing the polarity of the electric field as the device is addressed (the response of the liquid crystal layer is identical, regardless of the polarity of the applied field).

Displays for a small number of individual digits or fixed symbols (as in digital watches and pocket calculators) can be implemented with independent electrodes for each segment. In contrast, full alphanumeric or variable graphics displays are usually implemented with pixels arranged as a matrix consisting of electrically connected rows on one side of the LC layer and columns on the other side, which makes it possible to address each pixel at the intersections. The general method of matrix addressing consists of sequentially addressing one side of the matrix, for example by selecting the rows one-by-one and applying the picture information on the other side at the columns row-by-row. For details on the various matrix addressing schemes see Passive-matrix and active-matrix addressed LCDs



Reflective twisted nematic liquid display.

1. Polarizing filter film with a vertical axis to polarize light as it enters.
2. Glass substrate with ITO electrodes. The shapes of these electrodes will determine the shapes that will appear when the LCD is switched ON. Vertical ridges etched on the surface are smooth.
3. Twisted nematic liquid crystal.
4. Glass substrate with common electrode film (ITO) with horizontal ridges to line up with the horizontal filter.
5. Polarizing filter film with a horizontal axis to block/pass light.
6. Reflective surface to send light back to viewer. (In a backlit LCD, this layer is replaced with a light source.)

Illumination

Since LCD panels produce no light of their own, they require external light to produce a visible image. In a "transmissive" type of LCD, this light is provided at the back of the glass "stack" and is called the backlight. While passive-matrix displays are usually not backlit (e.g., calculators, wristwatches), active-matrix displays almost always are.

The common implementations of LCD backlight technology are:

CCFL: The LCD panel is lit either by two cathode fluorescents placed at opposite edges of the display or an array of parallel CCFLs behind larger displays. A diffuser then spreads the light out evenly across the whole display. For many years, this technology had been used almost exclusively. Unlike white LEDs, most CCFLs have an even-white spectral output resulting in better color gamut for the display. However, CCFLs are less energy efficient than LEDs and require a somewhat costly inverter to convert whatever DC voltage the device uses (usually 5 or 12 V) to ~1000 V needed to light a CCFL. The thickness of the inverter transformers also limit how thin the display can be made.

EL-WLED: The LCD panel is lit by a row of white LEDs placed at one or more edges of the screen. A light diffuser is then used to spread the light evenly across the whole display. As of 2012, this design is the most popular one in desktop computer monitors. It allows for the thinnest displays. Some LCD monitors using this technology have a feature called "Dynamic Contrast" where the backlight is dimmed to the brightest color that appears on the screen, allowing the 1000:1 contrast ratio of the LCD panel to be scaled to different light intensities, resulting in the "30000:1" contrast ratios seen in the advertising on some of these monitors. Since computer screen images usually have full white somewhere in the image, the backlight will usually be at full intensity, making this "feature" mostly a marketing gimmick.

WLED array: The LCD panel is lit by a full array of white LEDs placed behind a diffuser behind the panel. LCDs that use this implementation will usually have the ability to dim the LEDs in the dark areas of the image being displayed, effectively increasing the contrast ratio of the display. As of 2012, this design gets most of its use from upscale, larger-screen LCD televisions.

RGB-LED: Similar to the WLED array, except the panel is lit by a full array of RGB LEDs. While displays lit with white LEDs usually have a poorer color gamut than CCFL lit displays, panels lit with RGB LEDs have very wide color gamut’s. This implementation is most popular on professional graphics editing LCDs. As of 2012, LCDs in this category usually cost more than $1000.

4.5 FINGER PRINT RECOGNIZATION SYSTEM

FIG 4.4 FINGERPRINT SENSOR

A fingerprint is an impression of the friction ridges on all parts of the finger. A friction ridge is a raised portion of the epidermis on the palmar (palm) or digits (fingers and toes) or plantar (sole) skin, consisting of one or more connected ridge units of friction ridge skin. These are sometimes known as "epidermal ridges" which are caused by the underlying interface between the dermal papillae of the dermis and the interpapillary (rete) pegs of the epidermis. These epidermal ridges serve to amplify vibrations triggered when fingertips brush across an uneven surface, better transmitting the signals to sensory nerves involved in fine texture perception. The ridges assist in gripping rough surfaces, as well as smooth wet surfaces. Fingerprints may be deposited in natural secretions from the eccrine glands present in friction ridge skin (secretions consisting primarily of water) or they may be made by ink or other contaminants transferred from the peaks of friction skin ridges to a relatively smooth surface such as a fingerprint card. The term fingerprint normally refers to impressions transferred from the pad on the last joint of fingers and thumbs, though fingerprint cards also typically record portions of lower joint areas of the fingers (which are also used to make identifications).

4.5.1 FINGERPRINTS AS USED FOR IDENTIFICATION

Fingerprint identification (sometimes referred to as dactyloscopy or palm print identification is the process of comparing questioned and known friction skin ridge impressions (see Minutiae) from fingers or palms or even toes to determine if the impressions are from the same finger or palm. The flexibility of friction ridge skin means that no two finger or palm prints are ever exactly alike (never identical in every detail), even two impressions recorded immediately after each other. Fingerprint identification (also referred to as individualization) occurs when an expert (or an expert computer system operating under threshold scoring rules) determines that two friction ridge impressions originated from the same finger or palm (or toe, sole) to the exclusion of all others.

A known print is the intentional recording of the friction ridges, usually with black printers ink rolled across a contrasting white background, typically a white card. Friction ridges can also be recorded digitally using a technique called Live-Scan. A latent print is the chance reproduction of the friction ridges deposited on the surface of an item. Latent prints are often fragmentary and may require chemical methods, powder, or alternative light sources in order to be visualized.

When friction ridges come in contact with a surface that is receptive to a print, material on the ridges, such as perspiration, oil, grease, ink, etc. can be transferred to the item. The factors which affect friction ridge impressions are numerous, thereby requiring examiners to undergo extensive and objective study in order to be trained to competency. Pliability of the skin, deposition pressure, slippage, the matrix, the surface, and the development medium are just some of the various factors which can cause a latent print to appear differently from the known recording of the same friction ridges. Indeed, the conditions of friction ridge deposition are unique and never duplicated. This is another reason why extensive and objective study is necessary for examiners to achieve competency.

4.5.2 FINGERPRINT TYPES

LATENT PRINTS

Although the word latent means hidden or invisible, in modern usage for forensic science the term latent prints means any chance of accidental impression left by friction ridge skin on a surface, regardless of whether it is visible or invisible at the time of deposition. Electronic, chemical and physical processing techniques permit visualization of invisible latent print residue whether they are from natural secretions of the eccrine glands present on friction ridge skin (which produce palmar sweat, consisting primarily of water with various salts and organic compounds in solution), or whether the impression is in a contaminant such as motor oil, blood, paint, ink, etc. There are different types of fingerprint patterns such as an arch, tented arch, a loop, and a whorl. Each indicate what type of fingerprint it is.

Latent prints may exhibit only a small portion of the surface of the finger and may be smudged, distorted, overlapping, or any combination, depending on how they were deposited. For these reasons, latent prints are an "inevitable source of error in making comparisons," as they generally "contain less clarity, less content, and less undistorted information than a fingerprint taken under controlled conditions, and much, much less detail compared to the actual patterns of ridges and grooves of a finger."

PATENT PRINTS

These are friction ridge impressions of unknown origins which are obvious to the human eye and are caused by a transfer of foreign material on the finger, onto a surface. Because they are already visible, they need no enhancement, and are generally photographed instead of being lifted in the same manner as latent prints. An attempt to preserve the actual print is always made with numerous techniques; for later presentation in court. Finger deposits can include materials such as ink, dirt, or blood onto a surface.

PLASTICPRINTS

A plastic print is a friction ridge impression from a finger or palm (or toe/foot) deposited in a material that retains the shape of the ridge detail. Commonly encountered examples are melted candle wax, putty removed from the perimeter of window panes and thick grease deposits on car parts. Such prints are already visible and need no enhancement, but investigators must not overlook the potential that invisible latent prints deposited by accomplices may also be on such surfaces. After photographically recording such prints, attempts should be made to develop other non-plastic impressions deposited at natural finger/palm secretions (eccrine gland secretions) or contaminates.

CHAPTER 5

SOFTWARE

5.1 ARDUINO IDE

Atmel ARDUINO in 28-pin narrow DIP

The ARDUINO IDE is a modified Harvard architecture 8-bit RISC single chip microcontroller which was developed by Atmel in 1996. The ARDUINO IDE was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

Device overview

The ARDUINO IDE is a modified Harvard architecture machine where program and data are stored in separate physical memory systems that appear in different address spaces, but having the ability to read data items from program memory using special instructions.

Basic families

ARDUINO IDEs are generally classified into five broad groups:

- tiny ARDUINO IDE — the AT tiny series
- 0.5–8 kB program memory
- 6–32-pin package
- Limited peripheral set
- megaARDUINO IDE — the ATmega series
- 4–256 kB program memory
- 28–100-pin package

- Extended instruction set (Multiply instructions and instructions for handling larger program memories)
- Extensive peripheral set
- XMEGA — the ATxmega series
- 16–384 kB program memory
- 44–64–100-pin package (A4, A3, A1)
- Extended performance features, such as DMA, "Event System", and cryptography support.
- Extensive peripheral set with DACs
- Application-specific ARDUINO IDE
- megaARDUINO IDEs with special features not found on the other members of the ARDUINO IDE family, such as LCD controller, USB controller, advanced PWM, CAN etc.
- FPSLIC™ (ARDUINO IDE with FPGA)
- FPGA 5K to 40K gates
- SRAM for the ARDUINO IDE program code, unlike all other ARDUINO IDEs
- ARDUINO IDE core can run at up to 50 MHz [5]
- 32-bit ARDUINO IDEs

Main article: ARDUINO IDE32

In 2006 Atmel released microcontrollers based on the new, 32-bit, ARDUINO IDE32 architecture. They include SIMD and DSP instructions, along with other audio and video processing features. This 32-bit family of devices is intended to compete with the ARM based processors. The instruction set is similar to other RISC cores, but is not compatible with the original ARDUINO IDE or any of the various ARM cores.

Device architecture

Flash, EEPROM, and SRAM are all integrated onto a single chip, removing the need for external memory in most applications. Some devices have a parallel external bus option to allow adding additional data memory or memory-mapped devices. Almost all devices (except the smallest Tiny ARDUINO IDE chips) have serial interfaces, which can be used to connect larger serial EEPROMs or flash chips.

Program memory

Program instructions are stored in non-volatile flash memory. Although the MCUs are 8-bit, each instruction takes one or two 16-bit words.

The size of the program memory is usually indicated in the naming of the device itself (e.g., the ATmega64x line has 64 kB of flash while the ATmega32x line has 32 kB).

There is no provision for off-chip program memory; all code executed by the ARDUINO IDE core must reside in the on-chip flash. However, this limitation does not apply to the AT94 FPSLIC ARDUINO IDE/FPGA chips.

Internal data memory

The data address space consists of the register file, I/O registers, and SRAM.

Internal registers

Atmel ATxmega128A1 in 100-pin TQFP package

The ARDUINO IDEs have 32 single-byte registers and are classified as 8-bit RISC devices.

In most variants of the ARDUINO IDE architecture, the working registers are mapped in as the first 32 memory addresses (0000_{16}–$001F_{16}$) followed by the 64 I/O registers (0020_{16}–$005F_{16}$).

Actual SRAM starts after these register sections (address 0060_{16}). (Note that the I/O register space may be larger on some more extensive devices, in which case the memory mapped I/O registers will occupy a portion of the SRAM address space.)

Even though there are separate addressing schemes and optimized opcodes for register file and I/O register access, all can still be addressed and manipulated as if they were in SRAM.

In the XMEGA variant, the working register file is not mapped into the data address space; as such, it is not possible to treat any of the XMEGA's working registers as though they were SRAM. Instead, the I/O registers are mapped into the data address space starting at the very beginning of the address space. Additionally, the amount of data address space dedicated to I/O registers has grown substantially to 4096 bytes (0000_{16}–$0FFF_{16}$). As with previous generations, however, the fast I/O manipulation instructions can only reach the first 64 I/O register locations (the first 32 locations for bitwise instructions). Following the I/O registers, the XMEGA series sets aside a 4096-byte range of the data address space which can be used optionally for mapping the internal EEPROM to the data address space (1000_{16}–$1FFF_{16}$). The actual SRAM is located after these ranges, starting at 2000_{16}.

EEPROM

Almost all ARDUINO IDE microcontrollers have internal EEPROM for semi-permanent data storage. Like flash memory, EEPROM can maintain its contents when electrical power is removed.

In most variants of the ARDUINO IDE architecture, this internal EEPROM memory is not mapped into the MCU's addressable memory space. It can only be accessed the same way an external peripheral device is, using special pointer registers and read/write instructions which makes EEPROM access much slower than other internal RAM.

However, some devices in the Secure ARDUINO IDE (AT90SC) family [6] use a special EEPROM mapping to the data or program memory depending on the configuration. The XMEGA family also allows the EEPROM to be mapped into the data address space.

Since the number of writes to EEPROM is not unlimited — Atmel specifies 100,000 write cycles in their datasheets — a well-designed EEPROM write routine should compare the contents of an EEPROM address with desired contents and only perform an actual write if contents need to be changed.

Program execution

Atmel's ARDUINO IDEs have a two stage, single level pipeline design. This means the next machine instruction is fetched as the current one is executing. Most instructions take just one or two clock cycles, making ARDUINO IDEs relatively fast among the eight-bit microcontrollers.

The ARDUINO IDE family of processors were designed with the efficient execution of compiled C code in mind and has several built-in pointers for the task.

Instruction set

Main article: Atmel ARDUINO IDE instruction set

The ARDUINO IDE Instruction Set is more orthogonal than those of most eight-bit microcontrollers, in particular the 8051 clones and PIC microcontrollers with which ARDUINO IDE competes today. However, it is not completely regular:

- Pointer registers X, Y, and Z have addressing capabilities that are different from each other.
- Register locations R0 to R15 have different addressing capabilities than register locations R16 to R31.
- I/O ports 0 to 31 have different addressing capabilities than I/O ports 32 to 63.
- CLR affects flags, while SER does not, even though they are complementary instructions. CLR set all bits to zero and SER sets them to one. (Note that CLR is pseudo-op for EOR R, R; and SER is short for LDI R, $FF. Math operations such as EOR modify flags while moves/loads/stores/branches such as LDI do not.)
- Accessing read-only data stored in the program memory (flash) requires special LPM instructions; the flash bus is otherwise reserved for instruction memory.

Additionally, some chip-specific differences affect code generation. Code pointers (including return addresses on the stack) are two bytes long on chips with up to 128 Kbytes of flash memory, but three bytes long on larger chips; not all chips have hardware multipliers; chips with over 8 Kbytes of flash have branch and call instructions with longer ranges; and so forth.

The mostly-regular instruction set makes programming it using C (or even Ada) compilers fairly straightforward. GCC has included ARDUINO IDE support for quite some time, and that support is widely used. In fact, Atmel solicited input from major developers of compilers for small microcontrollers, to determine the instruction set features that were most useful in a compiler for high-level languages.

MCU speed

The ARDUINO IDE line can normally support clock speeds from 0-20 MHz, with some devices reaching 32 MHz’s Lower powered operation usually requires a reduced clock speed. All recent (Tiny, Mega, and Xmega, but not 90S) ARDUINO IDEs feature an on-chip oscillator, removing the need for external clocks or resonator circuitry. Some ARDUINO IDEs also have a system clock prescaler that can divide down the system clock by up to 1024. This prescaler can be reconfigured by software during run-time, allowing the clock speed to be optimized.

Since all operations (excluding literals) on registers R0 - R31 are single cycle, the ARDUINO IDE can achieve up to 1 MIPS per MHz, i.e., an 8 MHz processor can achieve up to 8 MIPS. Loads and stores to/from memory take 2 cycles, branching takes 2 cycles. Branches in the latest "3-byte PC" parts such as ATmega2560 are one cycle slower than on previous devices.

Development

ARDUINO IDEs have a large following due to the free and inexpensive development tools available, including reasonably priced development boards and free development software. The ARDUINO IDEs are sold under various names that share the same basic core but with different peripheral and memory combinations. Compatibility between chips in each family is fairly good, although I/O controller features may vary.

Features

Current ARDUINO IDEs offer a wide range of features:

- Multifunction, bi-directional general-purpose I/O ports with configurable, built-in pull-up resistors
- Multiple internal oscillators, including RC oscillator without external parts
- Internal, self-programmable instruction flash memory up to 256 kB (384 kB on XMega)
- In-system programmable using serial/parallel low-voltage proprietary interfaces or JTAG
- Optional boot code section with independent lock bits for protection
- On-chip debugging (OCD) support through JTAG or debugWIRE on most devices

- The JTAG signals (TMS, TDI, TDO, and TCK) are multiplexed on GPIOs. These pins can be configured to function as JTAG or GPIO depending on the setting of a fuse bit, which can be programmed via ISP or HVSP. By default, ARDUINO IDEs with JTAG come with the JTAG interface enabled.
- debugWIRE uses the /RESET pin as a bi-directional communication channel to access on-chip debug circuitry. It is present on devices with lower pin counts, as it only requires one pin.
- Internal data EEPROM up to 4 kB
- Internal SRAM up to 16 kB (32 kB on XMega)
- External 64 kB little endian data space on certain models, including the Mega8515 and Mega162.
- The external data space is overlaid with the internal data space, such that the full 64 kB address space does not appear on the external bus. An access to e.g., address 0100_{16} will access internal RAM, not the external bus.
- In certain members of the XMega series, the external data space has been enhanced to support both SRAM and SDRAM. As well, the data addressing modes have been expanded to allow up to 16 MB of data memory to be directly addressed.
- ARDUINO IDEs generally do not support executing code from external memory. Some ASSPs using the ARDUINO IDE core do support external program memory.
- 8-bit and 16-bit timers
- PWM output (some devices have an enhanced PWM peripheral which includes a dead-time generator)
- Input capture
- Analog comparator
- 10 or 12-bit A/D converters, with multiplex of up to 16 channels
- 12-bit D/A converters
- A variety of serial interfaces, including
- I^2C compatible Two-Wire Interface (TWI)

- Synchronous/asynchronous serial peripherals (UART/USART) (used with RS-232, RS-485, and more)
- Serial Peripheral Interface Bus (SPI)
- Universal Serial Interface (USI) for two or three-wire synchronous data transfer
- Brownout detection
- Watchdog timer (WDT)
- Multiple power-saving sleep modes
- Lighting and motor control (PWM-specific) controller models
- CAN controller support
- USB controller support
- Proper full-speed (12 Mbit/s) hardware & Hub controller with embedded ARDUINO IDE.
- Also, freely available low-speed (1.5 Mbit/s) (HID) bitbanging software emulations
- Ethernet controller support
- LCD controller support
- Low-voltage devices operating down to 1.8 V (to 0.7 V for parts with built-in DC–DC upconverter)
- picoPower devices
- DMA controllers and "event system" peripheral communication.
- Fast cryptography support for AES and DES

Programming interfaces

There are many means to load program code into an ARDUINO IDE chip. The methods to program ARDUINO IDE chips vary from ARDUINO IDE family to family.

ISP

The In-system programming (ISP) programming method is functionally performed through SPI, plus some twiddling of the Reset line. As long as the SPI pins of the ARDUINO IDE aren't connected to anything disruptive, the ARDUINO IDE chip can stay soldered on a PCB while reprogramming. All that's needed is a 6-pin connector and programming adapter. This is the most common way to develop with an ARDUINO IDE.

The Atmel ARDUINO IDE ISP mkII device connects to a computer's USB port and performs in-system programming using Atmel's software.

ARDUINO IDEDUDE (ARDUINO IDE Downloder UploaDEr) runs on Linux, FreeBSD, Windows, and Mac OS X, and supports a variety of in-system programming hardware, including Atmel ARDUINO IDE ISP mkII, Atmel JTAG ICE, older Atmel serial-port based programmers, and various third-party and "do-it-yourself" programmers.

PDI

The Program and Debug Interface (PDI) is an Atmel proprietary interface for external programming and on-chip debugging of XMEGA devices. The PDI supports high-speed programming of all non-volatile memory (NVM) spaces; flash, EEPROM, fuses, lock-bits and the User Signature Row. This is done by accessing the XMEGA NVM controller through the PDI interface, and executing NVM controller commands. The PDI is a 2-pin interface using the Reset pin for clock input (PDI_CLK) and a dedicated data pin (PDI_DATA) for input and output.

High voltage

High-voltage serial programming (hvsp) is mostly the backup mode on smaller ARDUINO IDEs. An 8-pin ARDUINO IDE package doesn't leave many unique signal combinations to place the ARDUINO IDE into a programming mode. A 12-volt signal, however, is something the ARDUINO IDE should only see during programming and never during normal operation.

Parallel

Parallel programming is considered the "final resort" and may be the only way to fix ARDUINO IDE chips with bad fuse settings. Parallel programming may be faster and beneficial when programming many ARDUINO IDE devices for production use.

Bootloader

Most ARDUINO IDE models can reserve a bootloader region, 256 B to 4 KB, where re-programming code can reside. At reset, the bootloader runs first, and does some user-programmed determination whether to re-program, or jump to the main application. The code can re-program through any interface available, it could read an encrypted binary through an Ethernet adapter like PXE. Atmel has application notes and code pertaining to many bus interfaces**.**

ROM

The AT90SC series of ARDUINO IDEs are available with a factory mask-ROM rather than flash for program memory.[14] Because of the large up-front cost and minimum order quantity, a mask-ROM is only cost-effective for high production runs.

aWire

aWire is a new one-wire debug interface available on the new UC3L ARDUINO IDE32 devices.

Debugging interfaces

The ARDUINO IDE offers several options for debugging, mostly involving on-chip debugging while the chip is in the target system.

Debug WIRE

debug WIRE™ is Atmel's solution for providing on-chip debug capabilities via a single microcontroller pin. It is particularly useful for lower pin count parts which cannot provide the four "spare" pins needed for JTAG. The JTAGICE mkII, mkIII and the ARDUINO IDE Dragon support debug WIRE. Debug WIRE was developed after the original JTAGICE release, and now clones support it.

JTAG

JTAG provides access to on-chip debugging functionality while the chip is running in the target system.[15] JTAG allows accessing internal memory and registers, setting breakpoints on code, and single-stepping execution to observe system behavior.

Atmel provides a series of JTAG adapters for the ARDUINO IDE:

- The JTAGICE 3 is the latest member of the JTAGICE family (JTAGICE mkIII). It supports JTAG, aWire, SPI, and PDI interfaces.
- The JTAGICE mkII replaces the JTAGICE, and is similarly priced. The JTAGICE mkII interfaces to the PC via USB, and supports both JTAG and the newer debugWIRE

interface. Numerous 3rd-party clones of the Atmel JTAGICE mkII device started shipping after Atmel released the communication protocol.[16]

- The ARDUINO IDE Dragon is a low-cost (approximately $50) substitute for the JTAGICE mkII for certain target parts. The ARDUINO IDE Dragon provides in-system serial programming, high-voltage serial programming and parallel programming, as well as JTAG or debugWIRE emulation for parts with 32 KB of program memory or less. ATMEL changed the debugging feature of ARDUINO IDE Dragon with the latest firmware of ARDUINO IDE STUDIO 4 - ARDUINO IDE STUDIO 5 and now it supports devices over 32KB of program memory.
- The JTAGICE adapter interfaces to the PC via a standard serial port. The JTAGICE has been End-Of-Lifed, though it is still supported in ARDUINO IDE Studio and other tools.

JTAG can also be used to perform a Boundary Scan test,[17] which tests the electrical connections between ARDUINO IDEs and other Boundary Scan capable chips in a system. Boundary scan is well-suited for a production line; the hobbyist is probably better off testing with a multimeter or oscilloscope.

Development tools and evaluation kits

Official Atmel ARDUINO IDE development tools and evaluation kits consists of a number of starter kits and debugging tools with support for most ARDUINO IDE devices:

STK600 starter kit

The STK600 starter kit and development system is an update to the STK500.[18] The STK600 uses a base board, a signal routing board, and a target board.

The base board is similar to the STK500, in that it provides a power supply, clock, in-system programming, an RS-232 port and a CAN (Controller Area Network, an automotive standard) port via DB9 connectors, and stake pins for all of the GPIO signals from the target device.

The target boards have ZIF sockets for DIP, SOIC, QFN, or QFP packages, depending on the board.

The signal routing board sits between the base board and the target board, and routes the signals to the proper pin on the device board. There are many different signal routing boards that could be used with a single target board, depending on what device is in the ZIF socket.

The STK600 interfaces with the PC via USB, leaving the RS-232 port available for the target microcontroller. A 4-pin header on the STK600 labeled 'RS-232 spare' can connect any TTL level USART port on the chip to the onboard MAX232 chip. The MAX232 is a TTL to RS-232 signal level converter to communicate with PC's. The pins are RX, TX, CTS, and RTS.

STK500 starter kit

The STK500 starter kit and development system features ISP and high voltage programming (HVP) for all ARDUINO IDE devices, either directly or through extension boards. The board is fitted with DIP sockets for all ARDUINO IDEs available in DIP packages.

STK500 Expansion Modules: Several expansion modules are available for the STK500 board:

- STK501 - Adds support for microcontrollers in 64-pin TQFP packages.
- STK502 - Adds support for LCD ARDUINO IDEs in 64-pin TQFP packages.
- STK503 - Adds support for microcontrollers in 100-pin TQFP packages.
- STK504 - Adds support for LCD ARDUINO IDEs in 100-pin TQFP packages.
- STK505 - Adds support for 14 and 20-pin ARDUINO IDEs.
- STK520 - Adds support for 14 and 20, and 32-pin microcontrollers from the AT90PWM and ATmega family.
- STK524 - Adds support for the ATmega32M1/C1 32-pin CAN/LIN/Motor Control family.
- STK525 - Adds support for the AT90USB microcontrollers in 64-pin TQFP packages.
- STK526 - Adds support for the AT90USB microcontrollers in 32-pin TQFP packages

STK200 starter kit

The STK200 starter kit and development system can use ARDUINO IDE chips via DIL-40/20/8 and features 4 MHz clock source, 8x Light-emitting diodes, 8x input buttons, RS-232 port, option for 32k SRAM and numerous general I/O. Programmed can be done with a dongle connected to the parallel-port and the ISP socket.

Software wise programs can be compiled with ARDUINO IDE-gcc, Simulated with Simu Arduino IDE, Downloaded with ARDUINO IDEdude/ARDUINO IDEice on BSD and Linux. Assembler is available with ARDUINO IDEa/tpasm. GNU debugger is available with ARDUINO IDE-gdb.

Support microcontrollers (according to the manual)

Chip	Flash size	EEPROM	SRAM	Frequency [MHz]	Package
AT90S1200	1k	64	0	12	PDIP-20
AT90S2313	2k	128	128	10	PDIP-20
AT90S/LS2323	2k	128	128	10	PDIP-8
AT90S/LS2343	2k	128	128	10	PDIP-8
AT90S4414	4k	256	256	8	PDIP-40
AT90S/LS4434	4k	256	256	8	PDIP-40
AT90S8515	8k	512	512	8	PDIP-40
AT90S/LS8535	8k	512	512	8	PDIP-40

ARDUINO IDE ISP and ARDUINO IDE ISP mkII

The ARDUINO IDE ISP and ARDUINO IDE ISP mkII are inexpensive tools allowing all ARDUINO IDEs to be programmed via ICSP.

The ARDUINO IDE ISP connects to a PC via a serial port, and draws power from the target system. The ARDUINO IDE ISP allows using either of the "standard" ICSP pinouts, either the 10-pin or 6-pin connector. The ARDUINO IDE ISP has been discontinued, replaced by the ARDUINO IDE ISP mkII.

The ARDUINO IDE ISP mkII connects to a PC via USB, and draws power from USB. LEDs visible through the translucent case indicate the state of target power.

ARDUINO IDE Dragon

ARDUINO IDE Dragon with ISP programming cable.

The Atmel Dragon is an inexpensive tool which connects to a PC via USB. The Dragon can program all ARDUINO IDEs via JTAG, HVP, PDI, or ICSP. The Dragon also allows debugging of all ARDUINO IDEs via JTAG, PDI, or DebugWire; a previous limitation to devices with 32 kB or less program memory has been removed in ARDUINO IDEstudio 4.18. The Dragon has a small prototype area which can accommodate an 8, 28, or 40-pin ARDUINO IDE, including connections to power and programming pins. There is no area for any additional circuitry, although this can be provided by a third-party product called the "Dragon Rider".

JTAGICE mkI

The JTAG In Circuit Emulator (JTAGICE) debugging tool supports on-chip debugging (OCD) of ARDUINO IDEs with a JTAG interface. The original JTAGICE mkI uses an RS-232 interface to a PC, and can only program ARDUINO IDEs with a JTAG interface. The JTAGICE mkI is no longer in production, however it has been replaced by the JTAGICE mkII.

JTAGICE mkII

The JTAGICE mkII debugging tool supports on-chip debugging (OCD) of ARDUINO IDEs with SPI, JTAG, PDI, and debug WIRE interfaces. The debug Wire interface enables debugging using only one pin (the Reset pin), allowing debugging of applications running on low pin-count microcontrollers.

The JTAGICE mkII connects using USB, but there is an alternate connection via serial port, which requires using a separate power supply. In addition to JTAG, the mkII supports ISP programming (using 6-pin or 10-pin adapters). Both the USB and serial links use a variant of the STK500 protocol.

Butterfly demo board

Atmel ATmega169 in 64-pad MLF package.

Main article: ARDUINO IDE Butterfly

The very popular ARDUINO IDE Butterfly demonstration board is a self-contained, battery-powered computer running the Atmel ARDUINO IDE ATmega169V microcontroller. It was built to show-off the ARDUINO IDE family, especially a new built-in LCD interface. The board includes the LCD screen, joystick, speaker, serial port, real time clock (RTC), flash memory chip, and both temperature and voltage sensors. Earlier versions of the ARDUINO IDE Butterfly also contained a CdS photoresistor; it is not present on Butterfly boards produced after June 2006 to allow RoHS compliance. The small board has a shirt pin on its back so it can be worn as a name badge.

The ARDUINO IDE Butterfly comes preloaded with software to demonstrate the capabilities of the microcontroller. Factory firmware can scroll your name, display the sensor readings, and show the time. The ARDUINO IDE Butterfly also has a piezo speaker that can be used to reproduce sounds and music.

The ARDUINO IDE Butterfly demonstrates LCD driving by running a 14-segment, six alpha-numeric character display. However, the LCD interface consumes many of the I/O pins.

The Butterfly's ATmega169 CPU is capable of speeds up to 8 MHz, however it is factory set by software to 2 MHz to preserve the button battery life. A pre-installed bootloader program allows the board to be re-programmed via a standard RS-232 serial plug with new programs that users can write with the free Atmel IDE tools.

AT90USBKey

This small board, about half the size of a business card, is priced at slightly more than an ARDUINO IDE Butterfly. It includes an AT90USB1287 with USB On-The-Go (OTG) support, 16 MB of DataFlash, LEDs, a small joystick, and a temperature sensor. The board includes software which lets it act as a USB Mass Storage device (its documentation is shipped on the DataFlash), a USB joystick, and more. To support the USB host capability, it must be operated from a battery; but when running as a USB peripheral, it only needs the power provided over USB.

Only the JTAG port uses conventional 2.54 mm pinout. All the other ARDUINO IDE I/O ports require more compact 1.27 mm headers.

The ARDUINO IDE Dragon can both program and debug since the 32 kb limitation was removed in ARDUINO IDE Studio 4.18, and the JTAGICE mkII is capable of both programming and debugging the processor. The processor can also be programmed through USB from a Windows or Linux host, using the USB "Device Firmware Update" protocols. Atmel ships proprietary (source code included but distribution restricted) example programs and a USB protocol stack with the device.

LUFA is a third-party free software (MIT license) USB protocol stack for the USBKey and other

Raven wireless kit

The RAVEN kit supports wireless development using Atmel's IEEE 802.15.4 chipsets, for ZigBee and other wireless stacks. It resembles a pair of wireless more-powerful Butterfly cards, plus a wireless USBKey; and costing about that much (under $US100). All these boards support JTAG-based development.

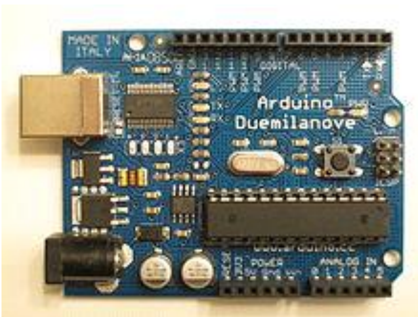
The kit includes two ARDUINO IDE Raven boards, each with 2.4 GHz transceiver supporting IEEE 802.15.4 (and a freely licensed ZigBee stack). The radios are driven with ATmega1284p processors, which are supported by a custom segmented LCD display driven by an ATmega3290p processor. Raven peripherals resemble the Butterfly: piezo speaker, DataFlash (bigger), external EEPROM, sensors, 32 kHz crystal for RTC, and so on. These are intended for use in developing remote sensor nodes, to control relays, or whatever is needed.

The USB stick uses an AT90USB1287 for connections to a USB host and to the 2.4 GHz wireless links. These are intended to monitor and control the remote nodes, relying on host power rather than local batteries.

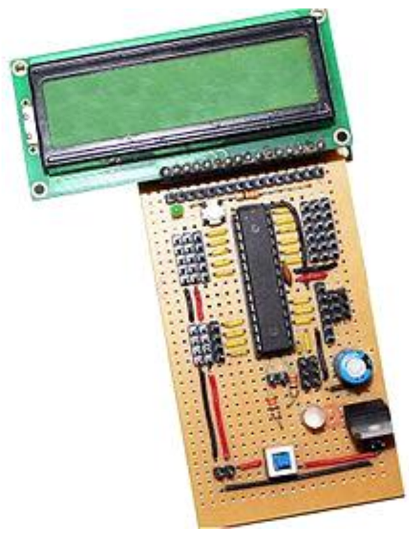
Third-party programmers

A wide variety of third-party programming and debugging tools are available for the ARDUINO IDE. These devices use various interfaces, including RS-232, PC parallel port, and USB. ARDUINO IDE Freaks has a comprehensive list.

Atmel ARDUINO IDE usage



Atmel ARDUINO IDE Atmega328 28-pin DIP on an Arduino Duemilanove board



Atmel ARDUINOIDE ARDUINO 28-pin DIP on a custom designed development board

ARDUINO IDEs have been used in various automotive applications such as security, safety, powertrain and entertainment systems. Atmel has recently launched a new publication

"Atmel Automotive Compilation" to help developers with automotive applications. Some current usages are in BMW, Daimler-Chrysler and TRW.

The Arduino physical computing platform is based on an ATmega328 microcontroller (ATmega168 or ARDUINO in older board versions than the Diecimila). The ATmega1280 and ATmega2560, with more pinout and memory capabilities, have also been employed to develop the Arduino Mega platform. Arduino boards can be used with its language and IDE, or with more conventional programming environments (C, assembler, etc.) as just standardized and widely available ARDUINO IDE platforms.

USB-based ARDUINO IDEs have been used in the Microsoft Xbox hand controllers. The link between the controllers and Xbox is USB.

Numerous companies produce ARDUINO IDE-based microcontroller boards intended for use by hobbyists, robot builders, experimenters and small system developers including: Cubloc, gnusb, BasicX, Oak Micros, ZX Microcontrollers, and myARDUINO IDE. There is also a large community of Arduino-compatible boards supporting similar users. Few hobbyists prefer making their own version of board from scratch.

Schneider Electric produces the M300 42 or and Motion Control Chip, incorporating an Atmel ARDUINO IDE Core and an Advanced Motion Controller for use in a variety of motion applications.[22]

FPGA clones

With the growing popularity of FPGAs among the open-source community, people have started developing open-source processors compatible with the ARDUINO IDE instruction set. The OpenCores website lists the following major ARDUINO IDE clone projects:

- pARDUINO IDE, written in VHDL, is aimed at creating the fastest and maximally featured ARDUINO IDE processor, by implementing techniques not found in the original ARDUINO IDE processor such as deeper pipelining.
- ARDUINO IDE_core, written in VHDL, is a clone aimed at being as close as possible to the ATmega103.
- NARDUINO IDEé, written in Verilog, implements all Classic Core instructions and is aimed at high performance and low resource usage. It does not support interrupts.

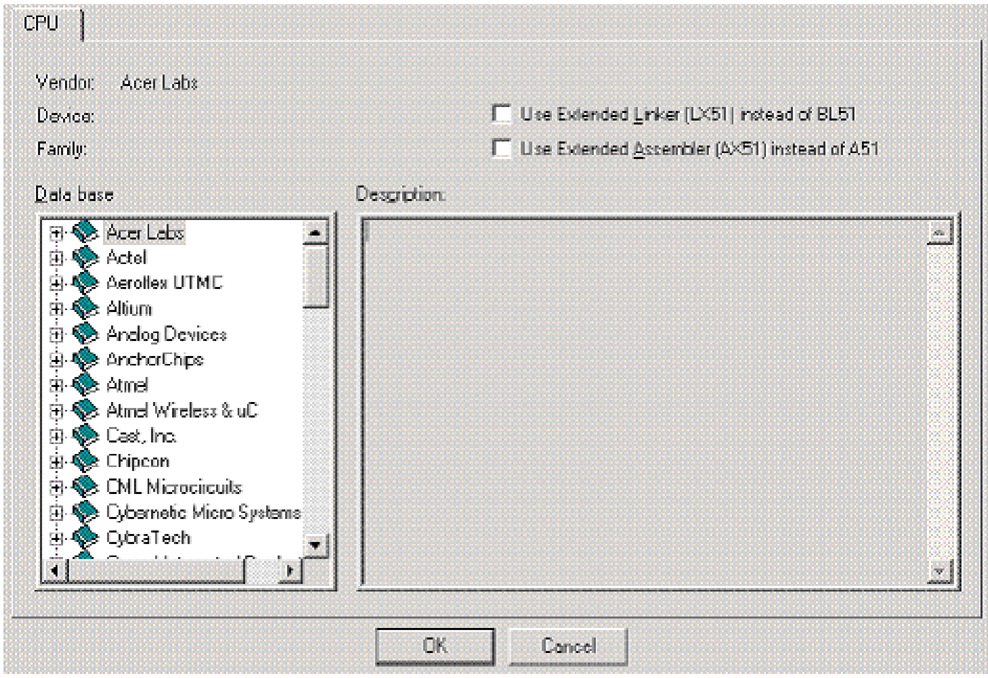


Fig 6.1: Window for choosing target device.

Next, Micro Vision must be instructed to generate a HEX file upon program compilation. A HEX file is a standard file format for storing executable code that is to be loaded onto the microcontroller. In the "Project Workspace" pane at the left, right–click on "Target 1" and select "Options for 'Target 1'". Under the "Output" tab of the resulting options dialog, ensure that both the "Create Executable" and "Create HEX File" options are checked. Then click "OK".

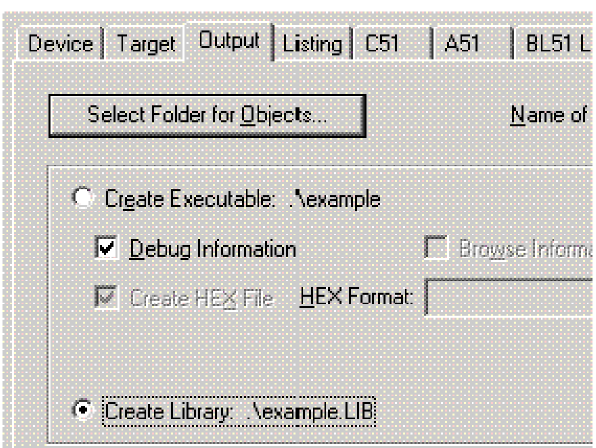


Fig 6.2: Project Options Dialog

Next, a file must be added to the project that will contain the project code. To do this, expand the "Target 1" heading, right–click on the "Source Group 1" folder, and select "Add files…" Create a new blank file (the file name should end in ".asm"), select it, and click "Add." The new file should now appear in the "Project Workspace" pane under the "Source Group 1" folder. Double-click on the newly created file to open it in the editor. All code for this lab will go in this file. To compile the program, first save all source files by clicking on the "Save All" button, and then click on the "Rebuild All Target Files" to compile the program as shown in the figure below. If any errors or warnings occur during compilation, they will be displayed in the output window at the bottom of the screen. All errors and warnings will reference the line and column number in which they occur along with a description of the problem so that they can be easily located. Note that only errors indicate that the compilation failed, warnings do not (though it is generally a good idea to look into them anyway).

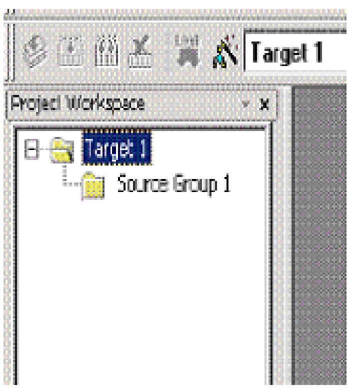


Fig 6.3: Project Workspace Pane

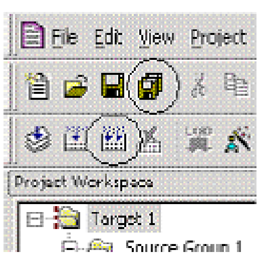


Fig 6.4: "Save All" and "Build All Target Files" Buttons

At the left side of the debugger window, a table is displayed containing several key parameters about the simulated microcontroller, most notably the elapsed time (circled in the figure below). Just above that, there are several buttons that control code execution. The "Run" button will cause the program to run continuously until a breakpoint is reached, whereas the "Step Into" button will execute the next line of code and then pause (the current position in the program is indicated by a yellow arrow to the left of the code).

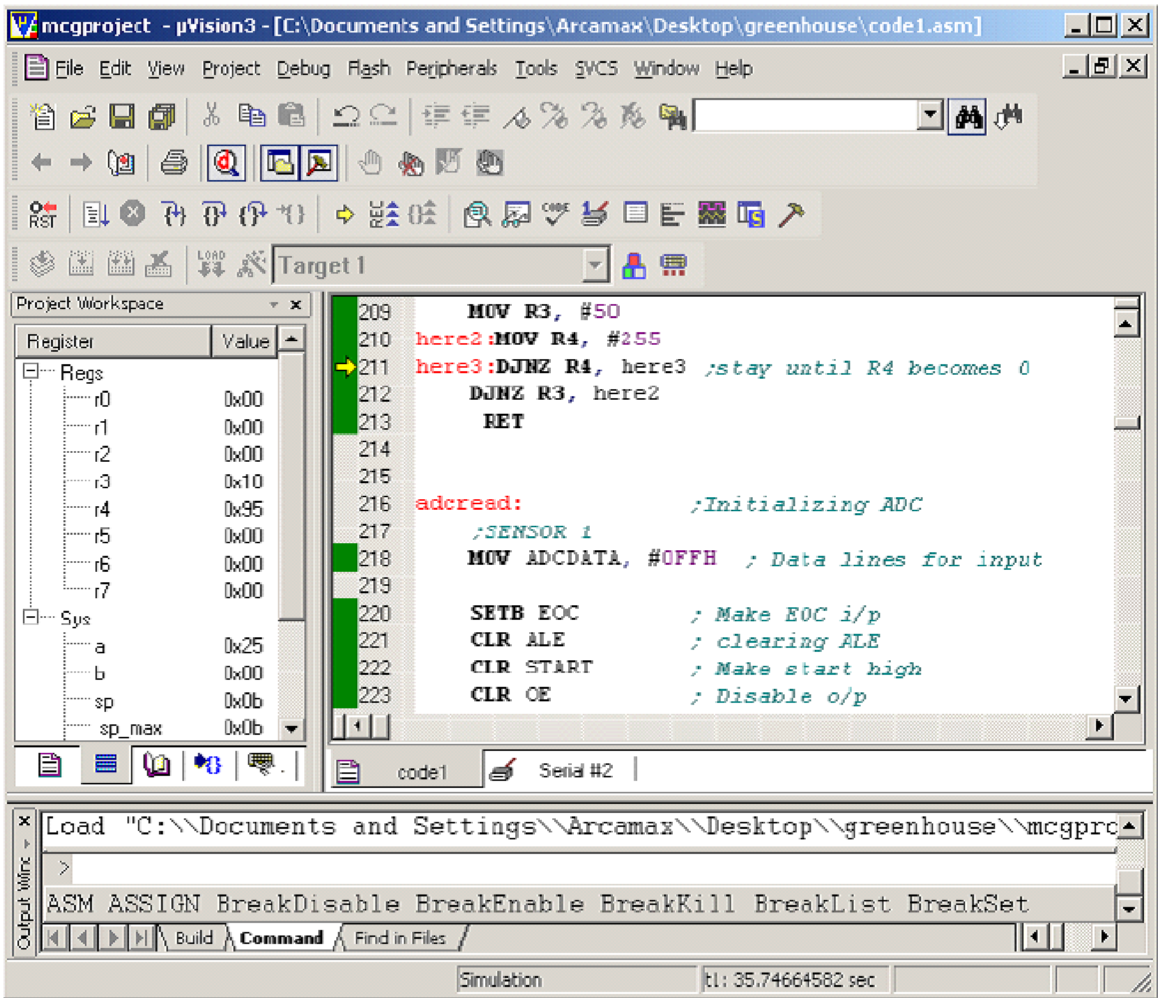


Fig 6.5: µVision3 Debugger window

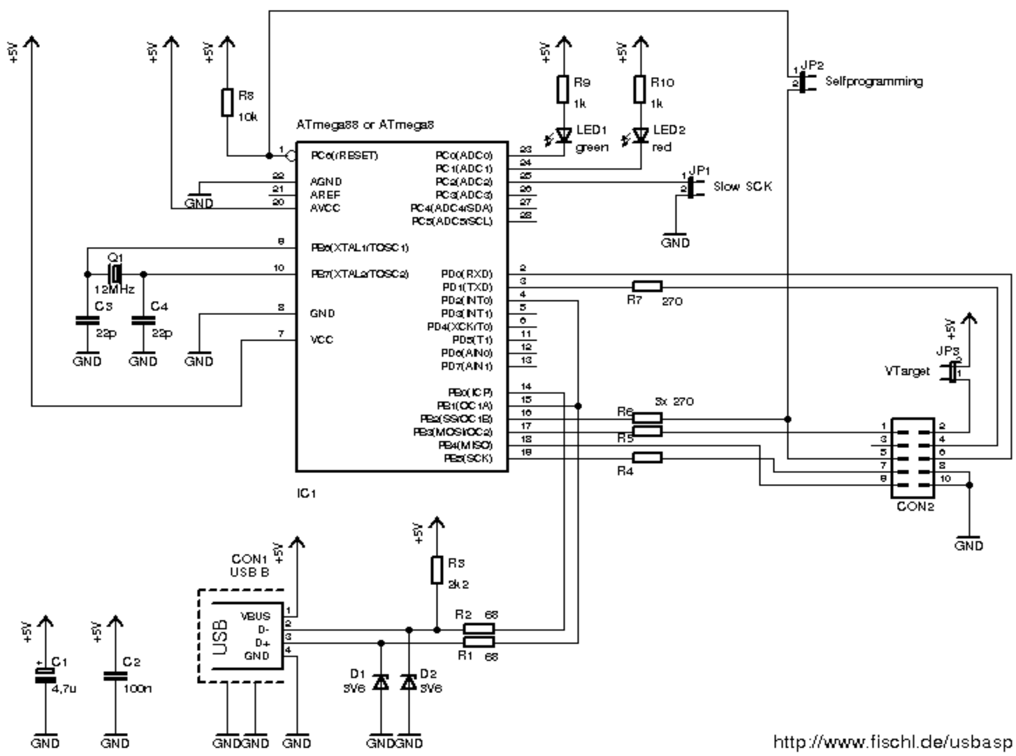
5.2 ARDUINO IDEDUDE ARDUINO IDE programmer

USBasp - USB programmer for Atmel ARDUINO IDE controllers: -

USBasp is a USB in-circuit programmer for Atmel ARDUINO IDE controllers. It simply consists of an ARDUINO and a couple of passive components. The programmer uses a firmware-only USB driver

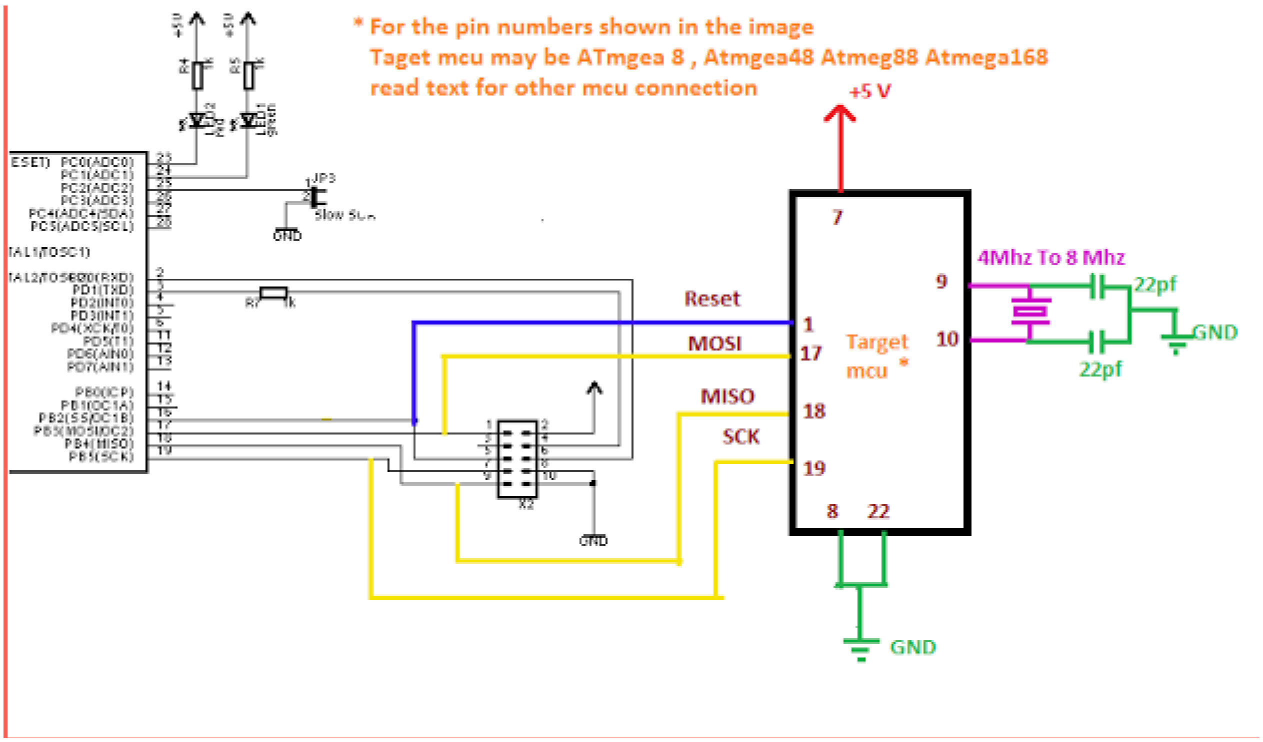
5.2.1 Features

- Flash Burner for ARDUINO IDE Series from ATMEL
- Communication - USB
- Auto Erase before writing and Auto Verify after writing
- Freeware ARDUINO IDE GCC C Compiler
- ISP Programming FRC Socket
- Connects through ARDUINO IDE DUDE
- Device Support



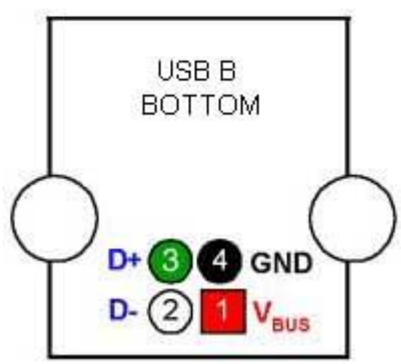
5.2.2 Connections to the target microcontroller: -

You can programmer any microcontroller by using this schematic, all you need to do is that find out the data sheet of that much you want to program and check the pin configuration. And the look for PIN MOSI MISO SCK and Reset, the connection will be as follows.



5.2.3 Connectors

USB B type female Connector



CHAPTER 6

CONCLUSION

Thus, here a voting system is discussed considering fingerprint matching process. Also, different techniques are analyzed. It is found that the fingerprint and face recognition-based voting system is best suitable in designing a proposed architecture. Hence, a proposed system will implement a voting system with privacy & security. The working of this model is very straightforward and very easy to understand. First, the fingerprint reader scans the fingerprint of the voter and sends the output to the microcontroller. The microcontroller then pairs the scanned data with the data in the database and retrieves the information about the voter. Now, the camera scans the face of the voter and checks whether it is similar to the face of the voter's face data that is paired with the fingerprint. There are many fraudulent and illegal activities that are happening in regards to the current voting process. With these problems in mind, the electronic voting machine is developed with fingerprint and facial recognition. This dual authentication system reduces the chances of the above-mentioned problems and so it has improved the security and efficiency of the voting process.

REFERENCES

[1] http://en.wikipedia.org/wiki/Biometrics.

[2] Amit Saxena, "IGNITION BASED ON FINGEPRINT RECOGNITION" Published in International Journal of Scientific Research and Management Studies (IJSRMS) Volume 2 Issue1.

[3] Prashant Kumar R. "TWO-WHEELER VEHICLE SECURITY SYSTEM" International Journal of Engineering Sciences & Emerging Technologies, Dec. 2013 Volume 6, Issue 3.

[4] H. Fakourfar and S. Belongie, "Fingerprint recognition system performance in the maritime environment", in Proc. Applications of Computer Vision, IEEE, 2009, pp1-5.

[5] http://www.biometricinfo.org/fingerprintrecognition.htm, "Biometrics Information Resource".

[6] Bhumi Bhatt, "Smart Vehicle Security System Using GSM & GPS" International Journal of Engineering and Computer Science Volume 4 Issue 6 June 2015.

[7] Roopam Arora "START-UP THE ENGINE USING FINGERPRINTING" International Journal of Computer Engineering and Applications, Volume IX, Issue X, Oct. 15.